TaskFlow: A Hierarchical, Priority-Weighted System for Project Progress and Performance Measurement

Aquino, Marl Aguiluz M.

Dela Cruz, Peter Marcus S.

Funclara, Edzel D.

Paghunasan, Jeroen Gil S.

Secretario, Kurt O.

Submitted In:

November 28, 2025

I. ABSTRACT

Traditional project management approach typically depend on spreadsheets and paper-based methods for planning, organizing, and tracking project progress. Nevertheless, these methods are frequently inefficient, susceptible to errors, and offer little insight into progress made in real-time. This study aims to develop TaskFlow, a web-based project management system aimed at fostering transparency, efficiency, accuracy, and accountability in project-based IT environments using a hierarchical, priority-weighted system for project progress and performance measurement. TaskFlow incorporates data-driven project monitoring, task coordination, and algorithmic performance evaluation to overcome the challenges faced by the traditional methods. A system development approach was utilized in the creation of TaskFlow, which involved the use of HTML5, CSS3, JavaScript, PHP, MySQL, and a weighted performance scoring model. The data collection method involved the simulation of real-world project scenarios typical of small to mid-sized organizations. The results demonstrate that TaskFlow has the potential to improve project management processes, especially in decision-making, worker performance assessment, and overall project tracking. Therefore, these findings demonstrate the potential of TaskFlow to serve as a reliable tool for modern, technology-driven project environments.

II. INTRODUCTION

Background of the Study

Currently, companies handle multiple projects that require efficient coordination, timely communication, and regular monitoring. Traditional manual methods, such as spreadsheets, paper documentation, and physical monitoring can lead to delays, inaccuracies, and difficulties in tracking project progress. These limitations can result in mismanagement of resources, poor collaboration among team members and an overall decline in productivity.

With the growing need for smooth operations, digital project management systems have become essential tools for organizing tasks, allocating resources, and monitoring project milestones in real time. However, existing solutions such as making a person monitor the project physically can be difficult and cost more. Therefore, this study focuses on developing a Project Management System (PMIS) designed to address these challenges and provide an effective platform for less hassle managing projects.

Statement of the Problem

This study aims to determine of the effectiveness of the TaskFlow based on the following questions:

1.1 To what extent does the adoption of TaskFlow improve overall project management efficiency based on phase-level granularity and weighted task completion progress?

1.2 What is the effect of TaskFlow usage on project manager performance as measured by project completion score and project time completion rate?

1.3 What is the effect of TaskFlow usage on worker performance based on task-level metrics (task status, task priority level, deadline adherence), worker assignment status (termination penalties), and project-level engagement (project count, completion rates, cross-project workload)?

Objectives of the Project

1.1 To determine the extent to which the adoption of TaskFlow improves overall project management efficiency based on phase-level granularity and weighted task completion progress.

1.2 To evaluate the effect of TaskFlow usage on project manager performance using metrics such as project completion score and project time completion rate.

1.3 To measure the effect of TaskFlow usage on worker performance based on task-level metrics (task status, task priority level, deadline adherence), worker assignment status (termination penalties), and project-level engagement (project count, completion rates, cross-project workload)?

Significance of the Study

The study aims for a structured data-driven approach for project monitoring, task coordination, and performance evaluation in project-based IT environments. Integrating timeline tracking, task visualization, and user performance metrics, TaskFlow enhances transparency, accuracy, and accountability in project management. The findings will enable project managers, team leaders, employees, and organizations to make better-informed decisions, assess progress in a more reliable way, and maintain clearer documentation of project outcomes. In the end, the system supports stronger stakeholder contributions through increased workflow efficiency, with guaranteed consistent processes, and leads to more organized and effective project delivery. These also serve as a valuable reference for future researchers on how to conduct workflow optimizations, digital project monitoring, and data-driven performance evaluations.

Scope of the Study

This study focuses on the development and evaluation of TaskFlow, a hierarchical and priority-weighted project management system designed for IT project environments. The system aims to enhance project monitoring, task coordination, and performance evaluation through structured dashboards, weighted progress indicators, and role-based access control.

Specifically, the scope includes the following:

1. The system provides functionalities for creating, assigning, and monitoring projects and their associated tasks, including progress tracking, priority weighting, and performance evaluation modules.
2. TaskFlow is designed primarily for Project Managers and Workers, with each user restricted to participating in only one active project at a time to ensure clearer workload distribution and accountability.
3. The study is conducted within an academic IT setting, simulating real-world project conditions typically found in small to mid-sized organizations.
4. The project was conceptualized and planned during the first semester of academic year 2025 - 2026, with system development and evaluation carried out from October 18 to November 28, 2025.

5. The study addresses inefficiencies found in traditional project tracking methods and introduces a customizable, user-friendly platform for structured performance measurement, task visualization, and milestone monitoring.
6. TaskFlow is developed using PHP 8.2 and MySQL/MariaDB, following a five-layer custom MVC architecture with performance calculators, role-based dashboards, and a fully normalized relational database to maintain structured workflows and accurate progress computation.

Limitations of the Study

This study is subject to the following limitations:

1. Users are limited to one active project at a time, simplifying tracking but not fully representing real-world multi-project workloads.
2. The system lacks integrated budget tracking and resource allocation monitoring functionality.
3. The accuracy of analytics and performance metrics relies on timely and correct data entry by users; delayed or inaccurate updates may affect results.
4. The system does not offer real-time communication tools such as messaging, file sharing, or document co-editing, reducing collaborative capabilities compared to full-scale PM platforms.

5. Features such as automated workload balancing, predictive analytics for project delays, and intelligent task reassignment are not included in the current version.
6. Factors like team dynamics, resource availability, managerial style, and unforeseen operational disruptions fall outside the system's scope and are not evaluated in this study.

III. REVIEW OF RELATED LITERATURE

Project Manager Information Systems (PMIS) are widely utilized to help managers plan, organize, and control projects. Several studies highlight the advantages of PMIS in terms of information flow, tracking, and decision-making accuracy. Findings from Maritim (2022) and Munyemana (2022) show improvements in coordination, reporting, and project alignment due to PMIS utilization. For example, in the context of public sector construction, Mutwiri and Muchelule (2022) noted that PMIS integration leads to increased transparency and performance, directly influencing improved project outcomes. Adebayo (2020) emphasized that digital tools not only streamline task tracking but also minimize delays, which enhances real-time visibility of project progress. Further, the innovations of AI-driven analytics, as discussed by Aminu (2024), demonstrate how advanced systems can refine progress measurement and resource allocation.

The quality of a Project Management Information System (PMIS) has a direct and significant influence on the decision-making effectiveness of project managers. According to Caniëls and Bakens (2012), when a PMIS provides high-quality information, this means that information is accurate, reliable, relevant, available, and easy to understand—project managers perceive it as more valuable and are more likely to rely on it when preparing project reports, allocating budgets, and managing resources. In a multi-project environment, where decision-making demands are higher, a high-quality PMIS greatly enhances decision-making by reducing the time required to make decisions, improving resource allocation, and strengthening overall monitoring and oversight of project activities.

Moreover, the research by Bloom (2015) indicates that a PMIS that display clear timelines, progress tracking, and visualization of tasks greatly enhances the well-being and efficiency of employees. These systems reduce clutter by displaying information in structured and easy-to-understand formats. They facilitate workflow by structuring tasks in ways that make it difficult for workers to feel overwhelmed. The work dashboard also displays speedy and accessible summaries of project status. Employees are able to visually track how much work has been completed on a progress chart. Due to this clarity, employees are often less stressed in managing their responsibilities. Therefore, order and good productivity occur in the workplace. Similarly, other researchers indicate that performance monitoring features of these systems facilitate decision-making among employees and managers (Kretschmer & Kude, 2021). These tools create a perfect project history which may guide future planning. Teams can adjust their strategies accordingly since there is reliable data behind real patterns and outcomes.

Project Management Information Systems (PMIS) play an increasingly important role in enhancing project efficiency, decision-making, and overall performance outcomes. Previous research demonstrates that PMIS adoption improves coordination, transparency, and project alignment across various industries, as discussed by Maritim (2022) and Mutwiri & Muchele (2022). Similarly, Adebayo (2020) and Aminu (2024) highlight how digital project management tools streamline task tracking, minimize delays, and provide real-time visibility of project progress. Concurrently, high-quality PMIS information also has been shown to strengthen decision-making effectiveness for project managers, particularly in multi-project environments (Caniëls & Bakens, 2012). Beyond managerial benefits, PMIS features such as progress dashboards, clear timelines, and structured task displays help reduce employee stress and promote productivity by improving task clarity and workflow organization (Bloom, 2015; Kretschmer & Kude, 2021).

This study aims to investigate how PMIS tools, particularly TaskFlow, contribute to project success provided the aforementioned advantages. By examining phase-level progress, schedule adherence, resource coordination, and task-level execution indicators, the study focuses on how TaskFlow usage affects project management efficiency, project manager performance, and worker task performance.

IV. SYSTEM ANALYSIS AND DESIGN

Requirements Analysis

Functional Requirements

1. Authentication & Authorization

    1.1. User Registration

    - System shall allow users to register with essential personal information and validate age older than 17
    - System shall send email confirmation link and store passwords securely (hashed)

    1.2. User Login & Session

    - System shall authenticate users via email/password and maintain sessions with CSRF protection
    - System shall support role-based access (Project Manager, Worker)

    1.3. Password Management

    - System shall support password reset via secure email links

2. User Management

2.1. User CRUD Operations

- System shall support creation, editing, and soft deletion of users
- System shall anonymize user data on deletion

2.2. Role-Based Permissions

- Project Manager shall manage workers, projects, tasks, and view all reports
- Worker shall view/update assigned tasks and view own performance

3. Project Management

3.1 Project Lifecycle

- System shall allow Project Managers to create, edit, and manage projects with: name, description, budget, dates, status
- System shall validate date constraints (completionDateTime must be later than the startDateTime, no past dates)
- System shall support statuses: pending, ongoing, completed, delayed, cancelled

3.2. Project Status Cascading

- System shall cascade cancellation: when project cancelled, all phases and tasks marked cancelled

3.3. Project Worker Assignment

- System shall allow managers to assign/remove workers from projects

4. Phase Management

4.1. Phase Management

- System shall allow creation and management of phases within projects
- System shall enforce hierarchical constraints: phase dates within project timeline
- System shall cascade status changes to child tasks

5. Task Management

5.1. Task Lifecycle

- System shall allow creation of tasks within phases with: name, description, dates, status, priority (low/medium/high)
- System shall validate date constraints and enforce phase timeline boundaries
- System shall set actualCompletionDateTime automatically when completed

5.2 Task Worker Assignment

- System shall assign multiple workers to tasks
- System shall track worker status per task

5.3. Task Filtering & Search

- System shall filter tasks by: status, priority, worker
- System shall support task search by name

6. Performance Calculation & Reporting

6.1. Project Progress Calculation

- System shall calculate hierarchical weighted progress:
- Weighted Progress = (Σ(Priority Weight × Status Completion) / Σ Priority Weights) × 100
- Phase-Weighted Progress = Σ(Phase Progress × Task Count) / Total Tasks
- Priority weights: high=3.0, medium=2.0, low=1.0
- Status completion: completed=100%, ongoing=50%, delayed=25%, pending/cancelled=0%

6.2. Worker Performance Calculation

- System shall calculate worker performance score (0-100):
- Task Score = Priority Weight × Status Multiplier × Time Multiplier

- Status multipliers: completed = 1.0, on going = 0.5, delayed = 0.3
- Time multipliers: early = 1.2, on time = 1.0, late = 0.8
- Priority weights: high = 5.0, medium = 3.0, low = 1.0

6.3. Project Manager Performance Calculation

- System shall calculate manager performance using dual metrics:
- Project Completion Score (status-based weights)
- Time Management Score (deadline adherence)
- Overall Score = (C_score × 0.45) + (T_score × 0.35)
- Status weights: completed = 1.0, ongoing = 0.6, delayed = 0.3, pending = 0.2, cancelled = -0.5
- Time multipliers: early = 1.3, on time = 1.0, late = 0.7, severely late = 0.4

6.4. Report Generation

- System shall generate reports with: progress metrics, status distribution, worker rankings, manager performance
- System shall visualize data using charts

7. Data Visualization

7.1 Performance Dashboards

- System shall display charts for: project progress, status distribution, worker performance comparisons
- System shall use color-coded badges for status and priority visualization

Non-Functional Requirements

1. Performance

1.1. Response Time

- API endpoints shall respond within 500ms for standard queries
- Performance calculations shall complete within 2 seconds for projects with < 1000 tasks

1.2. Database Optimization

- System shall use indexed queries on foreign keys (phaseId, projectId, workerId, taskId)
- System shall use optimized SQL with CASE statements for performance calculations

1.3. Scalability

- System shall support at least 100 concurrent users
- System shall handle projects with up to 1000 tasks without degradation

2. Security

2.1. Authentication & Authorization

- System shall hash passwords using secure algorithm (argon2id)
- System shall enforce role-based access control with permission hierarchy

2.2. CSRF Protection

- System shall include and validate CSRF tokens in all state-changing operations

2.3. Input Validation

- System shall validate all inputs server-side using prepared statements to prevent SQL injection
- System shall sanitize and validate data types per schema constraints

2.4. Data Privacy

- System shall anonymize user data on soft delete
- System shall use UUIDs (publicId) for external references

3. Reliability

3.1. Data Integrity

- System shall enforce referential integrity via foreign key constraints
- System shall use database triggers for automatic validations (age check, date consistency, status cascading)

3.2. Error Handling

- System shall log all exceptions to file system
- System shall handle database failures gracefully with transaction rollbacks

4. Maintainability

4.1. Code Architecture

- System shall follow MVC pattern with separation: Models (data), Controllers (pages), Endpoints (API), Validators (rules)
- System shall use path constants instead of relative paths

4.2. Code Quality

- System shall use PHP 8.2 type hints and enumerations
- System shall use abstract classes for common patterns (Model, Container, Validator)

4.3. Documentation

- System shall document performance calculation formulas

- System shall maintain technical documentation for algorithms and metrics

5. Compatibility

5.1. Technical Stack

- System shall run on PHP 8.2+, MySQL/MariaDB 10.4+
- System shall support modern browsers (Chrome, Firefox, Edge, Safari - latest 2 versions)

6. Operational

6.1. Configuration & Deployment

- System shall use environment variables for configuration (.env)
- System shall use Composer for PHP dependencies and PNPM for JavaScript dependencies

Database Design

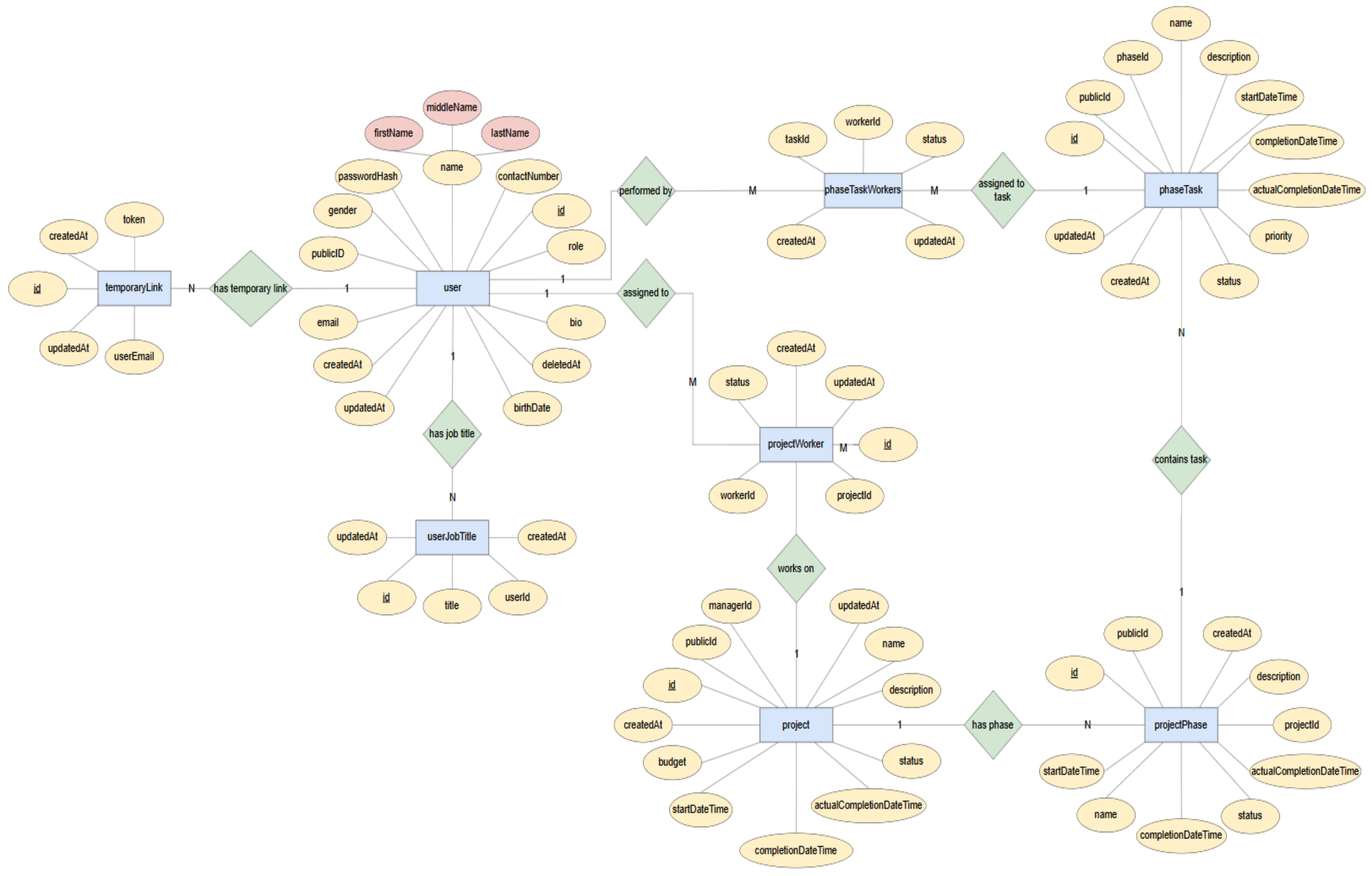


Figure 1. *ERD design of TaskFlow's database.*

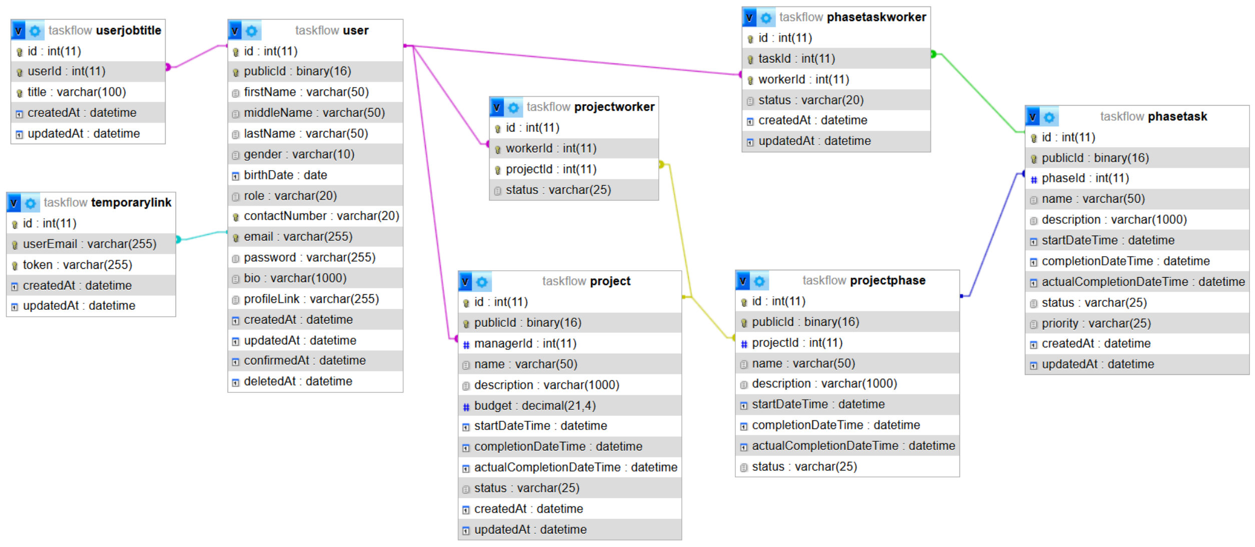


Figure 2. *Relational schema design of TaskFlow's database.*

The TaskFlow database follows a fully normalized relational structure designed to achieve First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF). The schema minimizes redundancy, maintains referential integrity through foreign keys, and prevents update anomalies by ensuring that each attribute is stored only once and derives solely from its appropriate key.

To satisfy 1NF, all tables store atomic values, avoid repeating groups, and use primary keys to uniquely identify each row. Core tables such as user, project, projectPhase, and phaseTask contain only single-valued attributes, while junction tables such as projectWorker and phaseTaskWorker appropriately model many-to-many relationships using composite primary keys.

The database meets 2NF by ensuring that every non-key attribute is fully dependent on the entire primary key. Since most tables use single-column primary keys, partial dependencies do not exist. In composite-key tables, attributes such as assignment status depend on the complete key pair rather than an individual component, preventing duplication of project, phase, or worker data.

Compliance with 3NF is achieved by removing transitive dependencies. Attributes such as user names, project managers, or priority levels are not duplicated across tables; instead, only their foreign keys are stored. Multi-valued attributes, such as job titles, are normalized into separate tables (e.g., userJobTitle) to preserve atomicity and avoid structural anomalies.

Several design decisions support normalization while maintaining performance.

These include:

- Hierarchical structure (Project → Phase → Task)
    - Stored strictly through foreign keys
    - Prevents redundancy, avoids transitive dependencies, and supports cascading updates
- Junction tables for many-to-many relationships
    - projectWorker and phaseTaskWorker ensure clean referential mapping

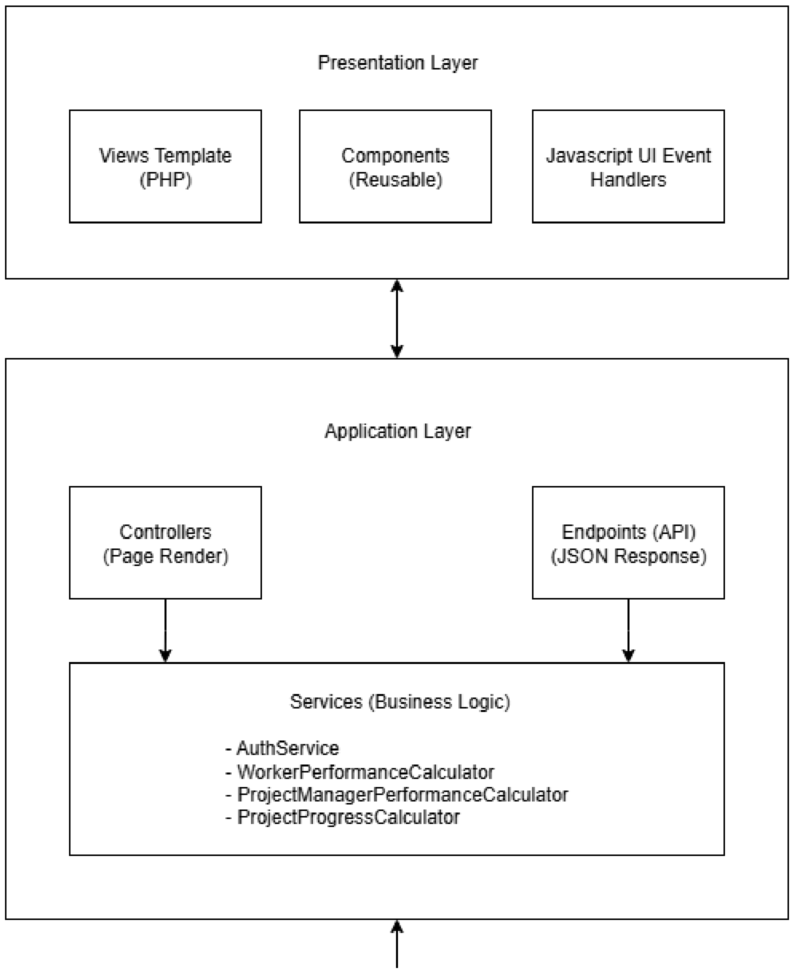
    - Avoids array-based IDs or repeated worker lists
- Separate table for job titles
    - Prevents comma-separated lists
    - Allows multiple titles per user through atomic rows
- Use of UUID public IDs
    - Auto-increment IDs optimize internal database operations
    - UUIDs protect sensitive sequential IDs from exposure
- Soft delete with anonymization
    - Maintains integrity of historical records
    - Replaces personal data rather than removing linked rows

Finally, foreign key constraints, unique indexes, CHECK constraints, events, and database triggers ensure persistent data accuracy. Cascading rules prevent orphaned rows, CHECK constraints validate enumerated values (status, priority), and events and triggers enforce business rules such as completion timestamps and status propagation across tasks, phases, and projects. Overall, TaskFlow's database maintains a strong balance between normalized design and practical performance optimization.

System Architecture

TaskFlow implements a five-layer MVC architecture that clearly distinguishes between API operations (Endpoints) and page rendering (Controllers). The presentation layer utilizes PHP templates with reusable components and modular JavaScript event handlers, while the application layer utilizes controllers for rendering HTML pages and Endpoints for JSON API responses. With specialised calculators (WorkerPerformanceCalculator, ProjectManagerPerformanceCalculator, ProjectProgressCalculator) that apply documented mathematical formulas, business logic is implemented in a dedicated service layer inside the application layer. Type-safe Entities for data representation, Containers for collections, Enumerations for fixed value sets (WorkStatus, TaskPriority, Role), and Validators for input rules comprise the domain layer. In order to prevent SQL injection and maintain a clear separation of concerns, the data access layer employs Model classes that execute pure CRUD operations using prepared statements without any business logic.

The database layer utilizes MySQL/MariaDB with a normalised relational schema that is enforced by foreign key constraints, CHECK constraints for enum validation, and database triggers for automated operations such as data anonymisation on user soft deletion, date consistency validation, status cascading, and automatic timestamp updates. The system makes use of path constants for maintainable file references, JSON-based routing configuration for flexible URL mapping, session-based authentication, role-based authorisation with permission hierarchies, CSRF token protection, and UUID-based public IDs to hide internal database structures. Strict layer boundaries are maintained for testability and long-term maintainability, and the architecture places a high priority on hierarchical data management with weighted performance metrics at each level. This allows for thorough progress tracking and performance evaluation.


Presentation Layer
Views Template (PHP)
Components (Reusable)
Javascript UI Event Handlers
Application Layer
Controllers (Page Render)
Endpoints (API) (JSON Response)
Services (Business Logic)
- AuthService
- WorkerPerformanceCalculator
- ProjectManagerPerformanceCalculator
- ProjectProgressCalculator

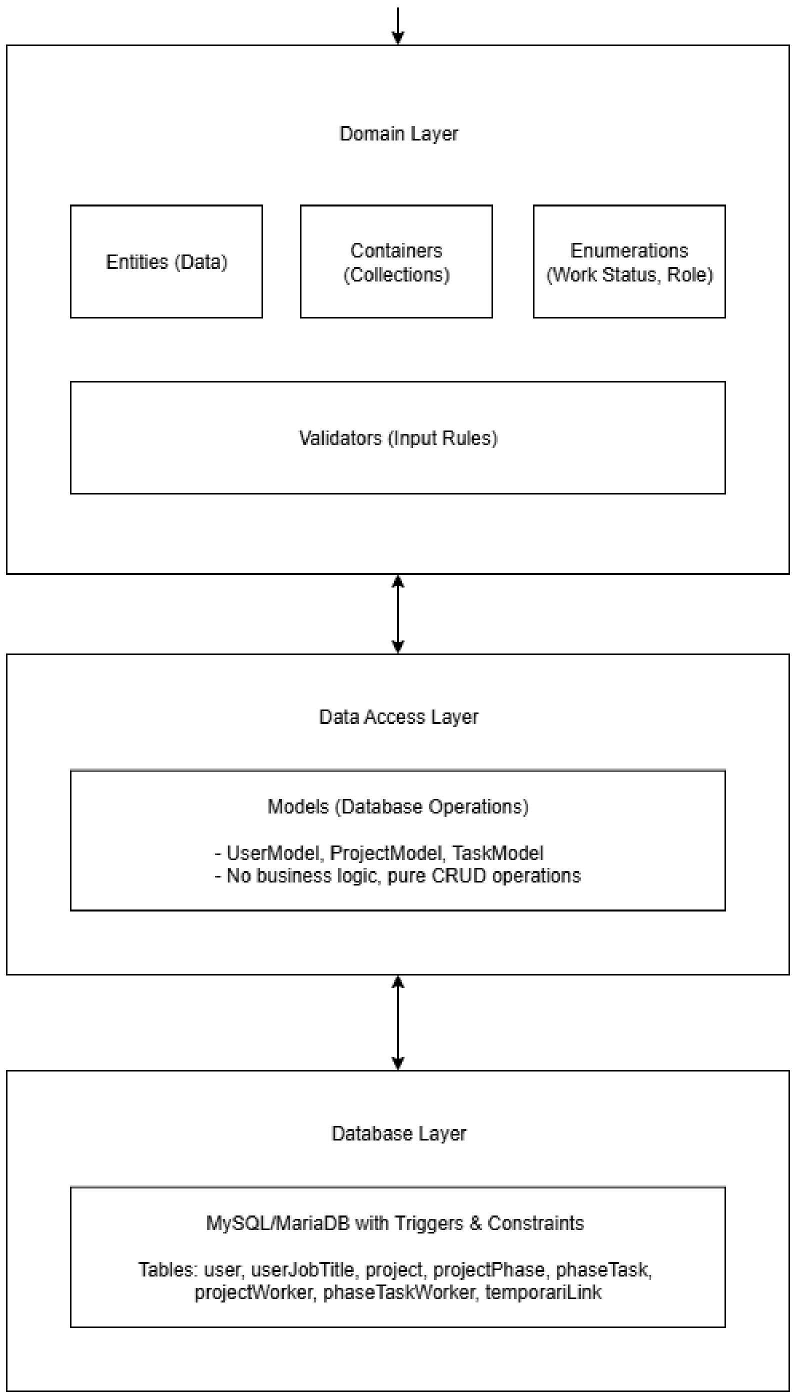
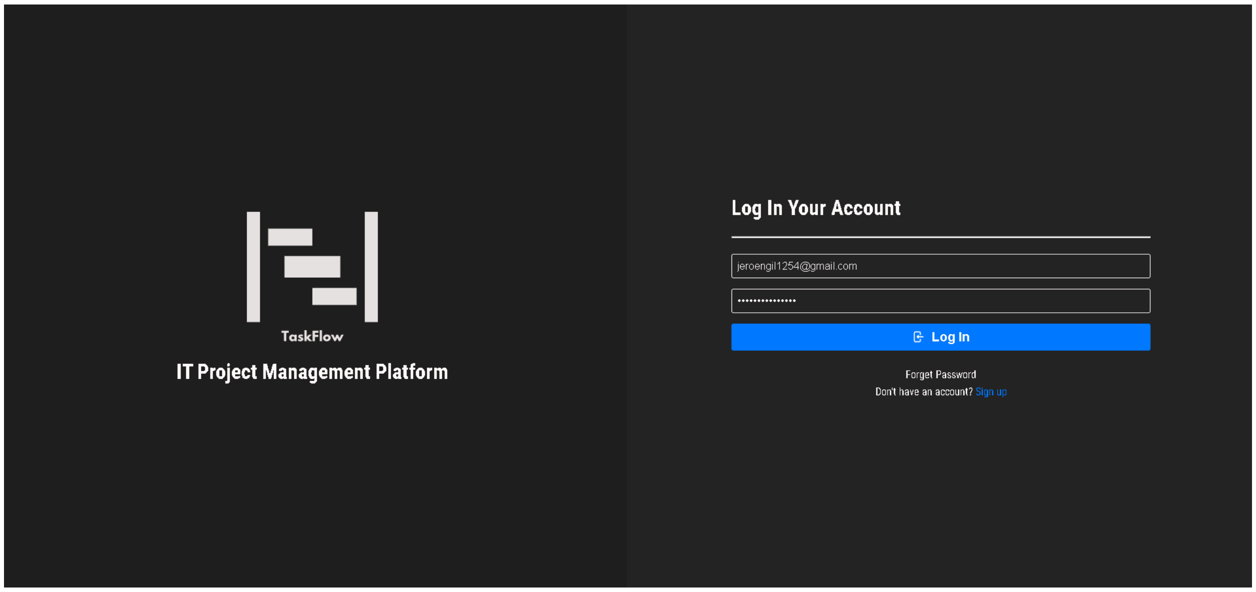

Domain Layer
Entities (Data)
Containers (Collections)
Enumerations (Work Status, Role)
Validators (Input Rules)
Data Access Layer
Models (Database Operations)
- UserModel, ProjectModel, TaskModel
- No business logic, pure CRUD operations
Database Layer
MySQL/MariaDB with Triggers & Constraints
Tables: user, userJobTitle, project, projectPhase, phaseTask, projectWorker, phaseTaskWorker, temporariLink

Figure 3*. TaskFlow's five-layer MVC architecture with Endpoint layer.*

Interface Design


TaskFlow
IT Project Management Platform

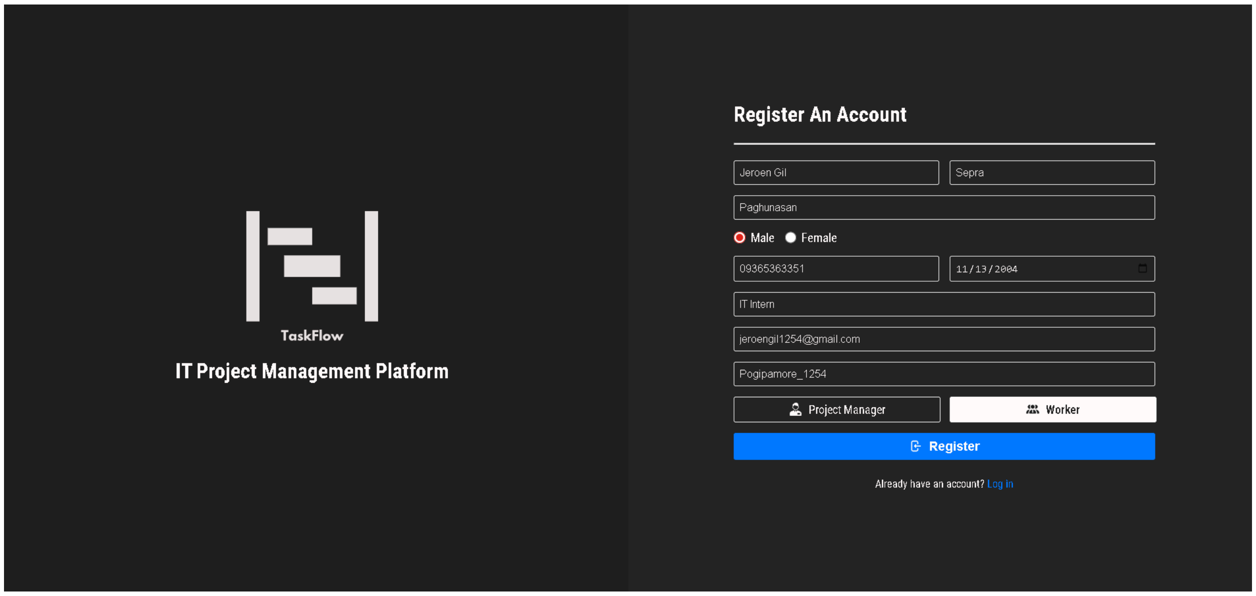
Log In Your Account

jeroengil1254@gmail.com

•••••••••••••••

Log In
Forget Password
Don't have an account? Sign up

Figure 4. *Login Interface.*


TaskFlow
IT Project Management Platform

Register An Account

Jeroen Gil | Sepra

Paghunasan

[✓ Male] Female

09365363351 | 11/13/2004

IT Intern

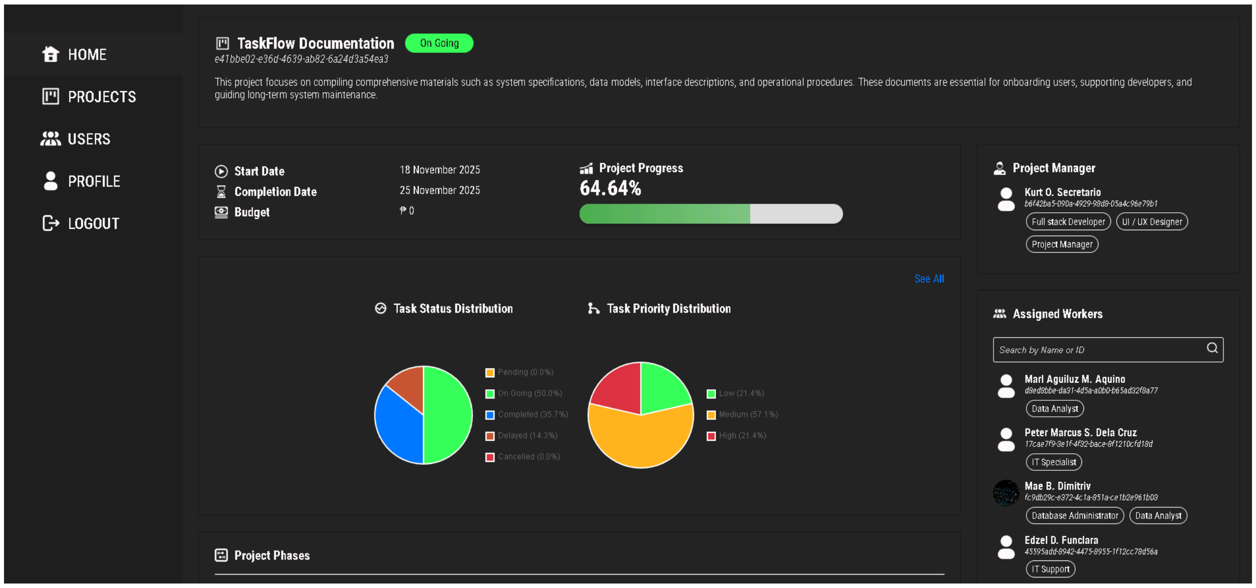
jeroengil1254@gmail.com

Pogipamore_1254

Project Manager [✓ Worker]

Register
Already have an account? Log in

Figure 5. *Register Interface.*



TaskFlow Documentation On Going
e41bbe02-e36d-4639-ab82-6a24d3a54ea3

This project focuses on compiling comprehensive materials such as system specifications, data models, interface descriptions, and operational procedures. These documents are essential for onboarding users, supporting developers, and guiding long-term system maintenance.

Start Date	18 November 2025
Completion Date	25 November 2025
Budget	₱ 0

Project Progress
64.64%

Project Manager
Kurt O. Secretario
b6f42ba5-090a-4929-98d8-05a4c96e79b1
Full stack Developer · UI / UX Designer · Project Manager

See All

Task Status Distribution
- 🟧 Pending (0.0%)
- 🟩 On Going (50.0%)
- 🟦 Completed (35.7%)
- 🟫 Delayed (14.3%)
- 🟥 Cancelled (0.0%)

Task Priority Distribution
- 🟩 Low (21.4%)
- 🟧 Medium (57.1%)
- 🟥 High (21.4%)

Project Phases

Assigned Workers
Search by Name or ID

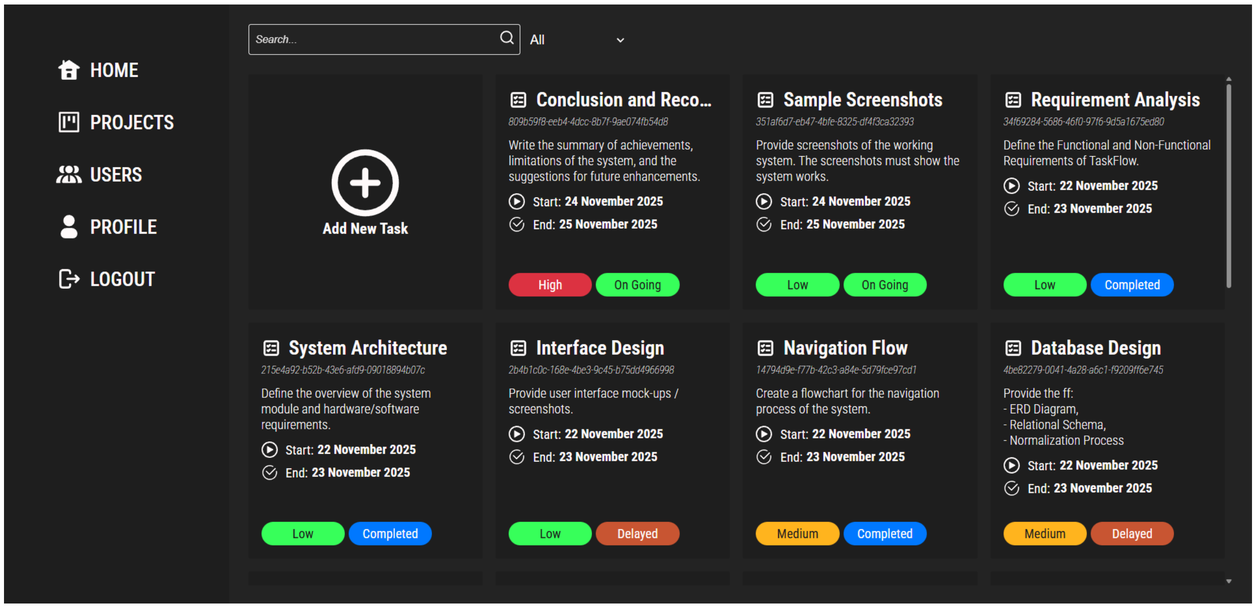
Marl Aguiluz M. Aquino
d8ed8bbe-da31-4d5a-a0b0-b65ad32f8a77
Data Analyst

Peter Marcus S. Dela Cruz
17cae7f9-3e1f-4f32-bace-8f1210cfd18d
IT Specialist

Mae B. Dimitriv
fc9db29c-e372-4c1a-851a-ce1b2e961b03
Database Administrator · Data Analyst

Edzel D. Funclara
45595add-8942-4475-8955-1f12cc78d56a
IT Support

Figure 6. *Home Interface.*



Conclusion and Reco...
809b59f8-eeb4-4dcc-8b7f-9ae074fb54d8
Write the summary of achievements, limitations of the system, and the suggestions for future enhancements.
Start: **24 November 2025**
End: **25 November 2025**
High · On Going

Sample Screenshots
351af6d7-eb47-4bfe-8325-df4f3ca32393
Provide screenshots of the working system. The screenshots must show the system works.
Start: **24 November 2025**
End: **25 November 2025**
Low · On Going

Requirement Analysis
34f69284-5686-46f0-97f6-9d5a1675ed80
Define the Functional and Non-Functional Requirements of TaskFlow.
Start: **22 November 2025**
End: **23 November 2025**
Low · Completed

System Architecture
215e4a92-b52b-43e6-afd9-09018894b07c
Define the overview of the system module and hardware/software requirements.
Start: **22 November 2025**
End: **23 November 2025**
Low · Completed

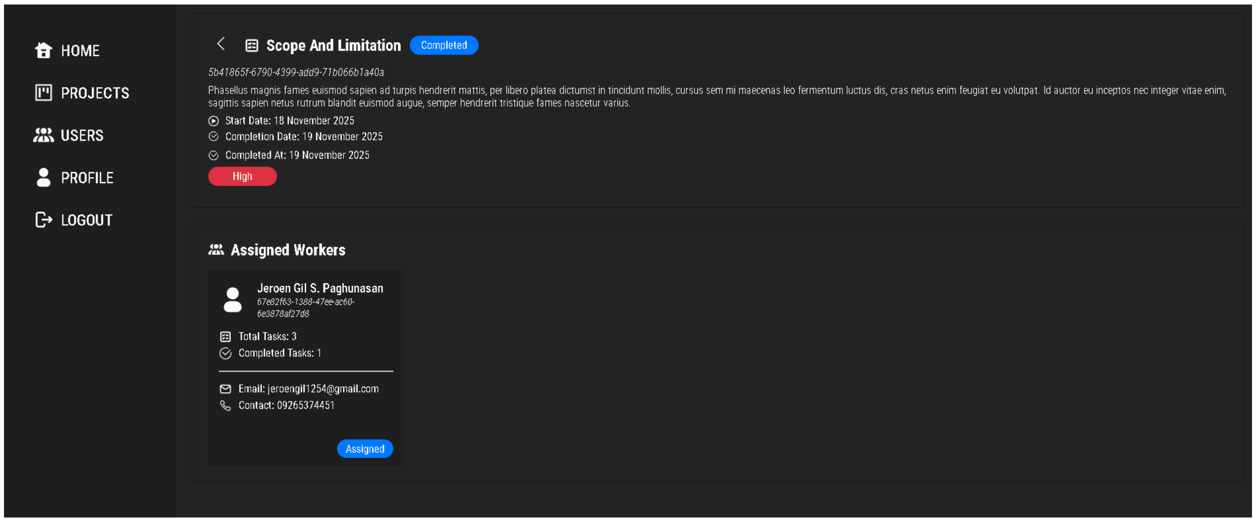
Interface Design
2b4b1c0c-168e-4be3-9c45-b75dd4966998
Provide user interface mock-ups / screenshots.
Start: **22 November 2025**
End: **23 November 2025**
Low · Delayed

Navigation Flow
14794d9e-f77b-42c3-a84e-5d79fce97cd1
Create a flowchart for the navigation process of the system.
Start: **22 November 2025**
End: **23 November 2025**
Medium · Completed

Database Design
4be82279-0041-4a28-a6c1-f9209ff6e745
Provide the ff:
- ERD Diagram,
- Relational Schema,
- Normalization Process

Start: **22 November 2025**
End: **23 November 2025**
Medium · Delayed

Figure 7. *Tasks List Interface.*



Scope And Limitation Completed

5b41865f-6790-4399-add9-71b066b1a40a

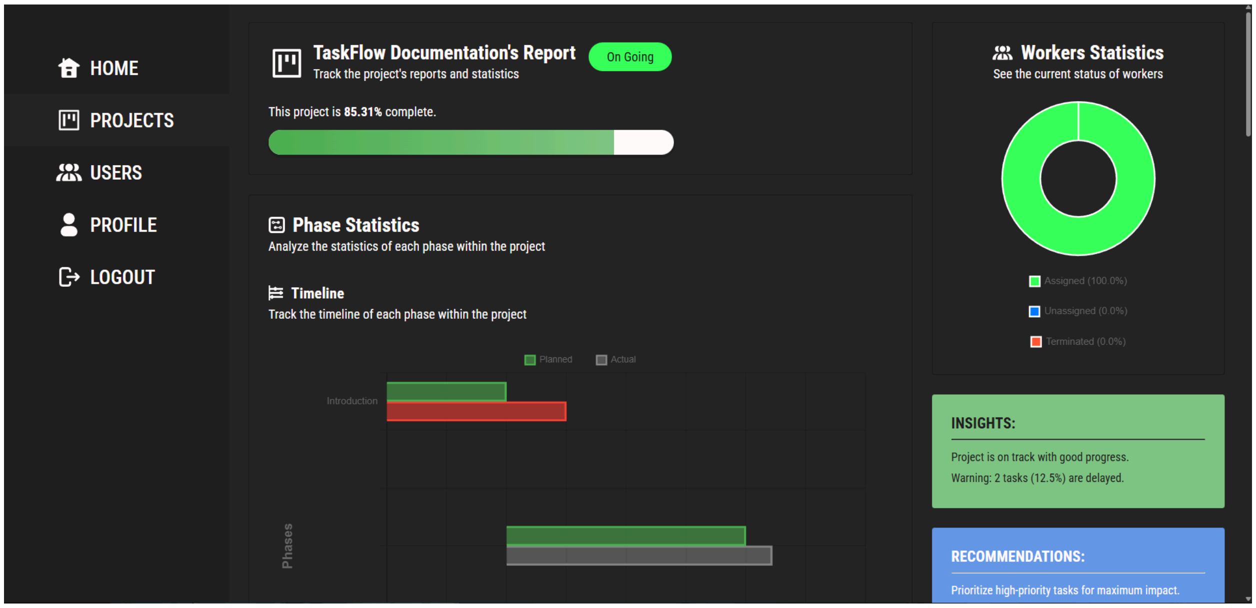
Phasellus magnis fames euismod sapien ad turpis hendrerit mattis, per libero platea dictumst in tincidunt mollis, cursus sem mi maecenas leo fermentum luctus dis, cras netus enim feugiat eu volutpat. Id auctor eu inceptos nec integer vitae enim, sagittis sapien netus rutrum blandit euismod augue, semper hendrerit tristique fames nascetur varius.

Start Date: 18 November 2025
Completion Date: 19 November 2025
Completed At: 19 November 2025
High

Assigned Workers

Jeroen Gil S. Paghunasan
67e82f63-1388-47ee-ac60-6e3878af27d8
Total Tasks: 3
Completed Tasks: 1
Email: jeroengil1254@gmail.com
Contact: 09265374451
Assigned

Figure 8. *Task Information Interface.*



TaskFlow Documentation's Report On Going
Track the project's reports and statistics

This project is **85.31%** complete.

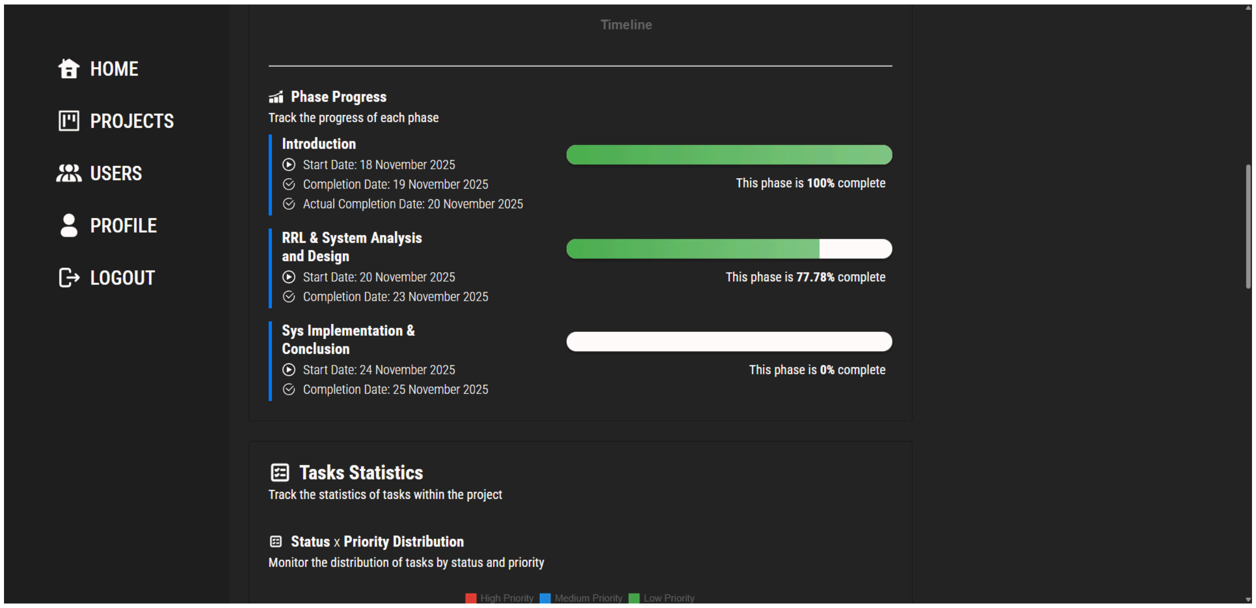
Phase Statistics
Analyze the statistics of each phase within the project

Timeline
Track the timeline of each phase within the project

Workers Statistics
See the current status of workers

INSIGHTS:
Project is on track with good progress.
Warning: 2 tasks (12.5%) are delayed.

RECOMMENDATIONS:
Prioritize high-priority tasks for maximum impact.



Phase Progress
Track the progress of each phase

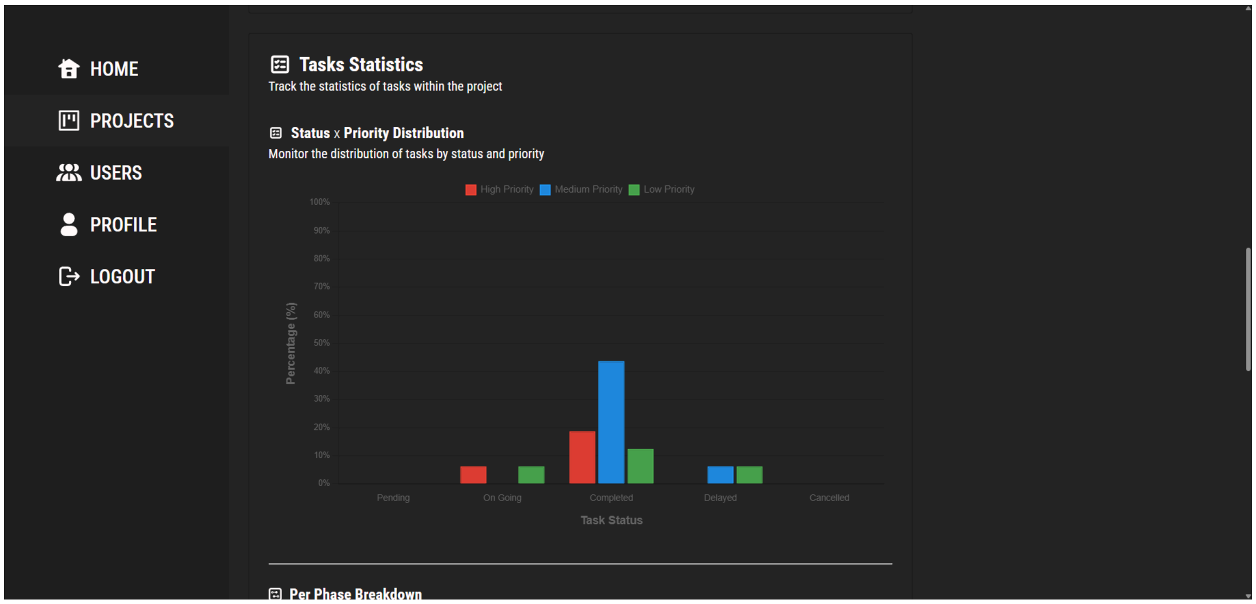
Introduction
Start Date: 18 November 2025
Completion Date: 19 November 2025
Actual Completion Date: 20 November 2025
This phase is **100%** complete

RRL & System Analysis and Design
Start Date: 20 November 2025
Completion Date: 23 November 2025
This phase is **77.78%** complete

Sys Implementation & Conclusion
Start Date: 24 November 2025
Completion Date: 25 November 2025
This phase is **0%** complete

Tasks Statistics
Track the statistics of tasks within the project

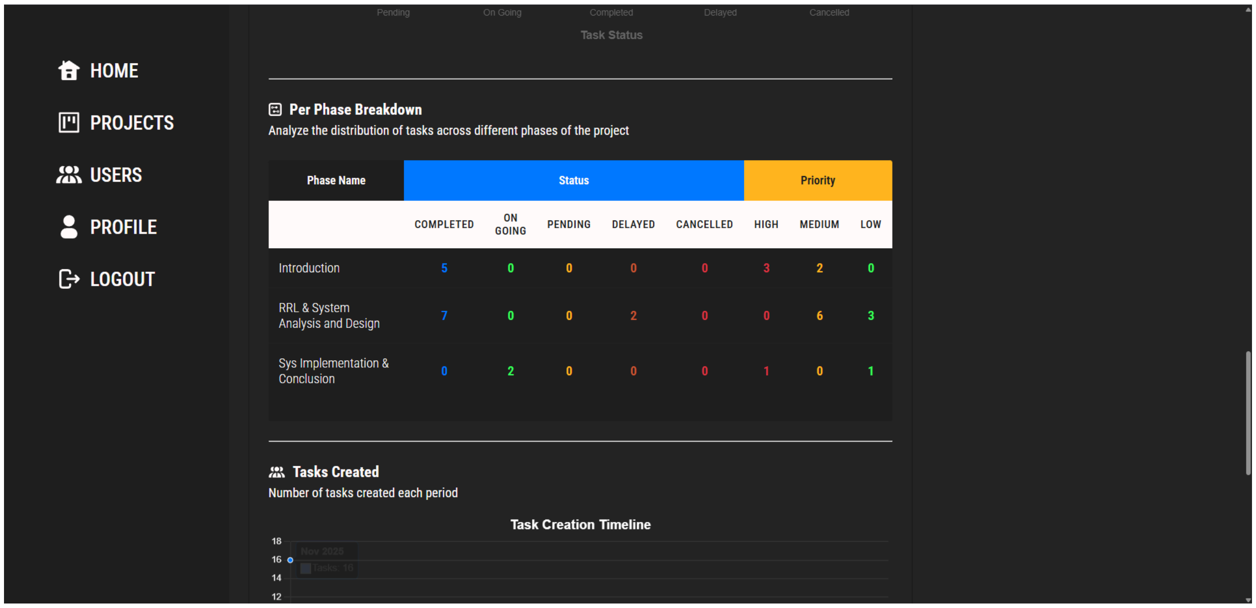
Status x **Priority Distribution**
Monitor the distribution of tasks by status and priority



Tasks Statistics
Track the statistics of tasks within the project

Status x **Priority Distribution**
Monitor the distribution of tasks by status and priority

Per Phase Breakdown



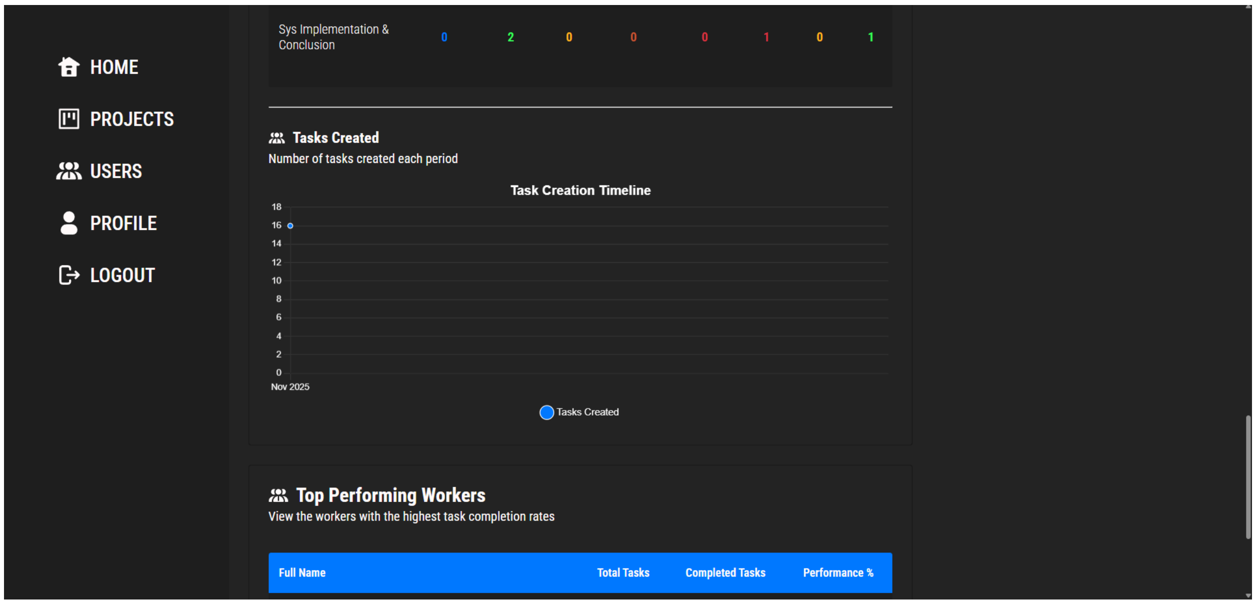
Per Phase Breakdown

Analyze the distribution of tasks across different phases of the project

Phase Name	Status					Priority		
	COMPLETED	ON GOING	PENDING	DELAYED	CANCELLED	HIGH	MEDIUM	LOW
Introduction	5	0	0	0	0	3	2	0
RRL & System Analysis and Design	7	0	0	2	0	0	6	3
Sys Implementation & Conclusion	0	2	0	0	0	1	0	1

Tasks Created

Number of tasks created each period



Sys Implementation & Conclusion	0	2	0	0	0	1	0	1

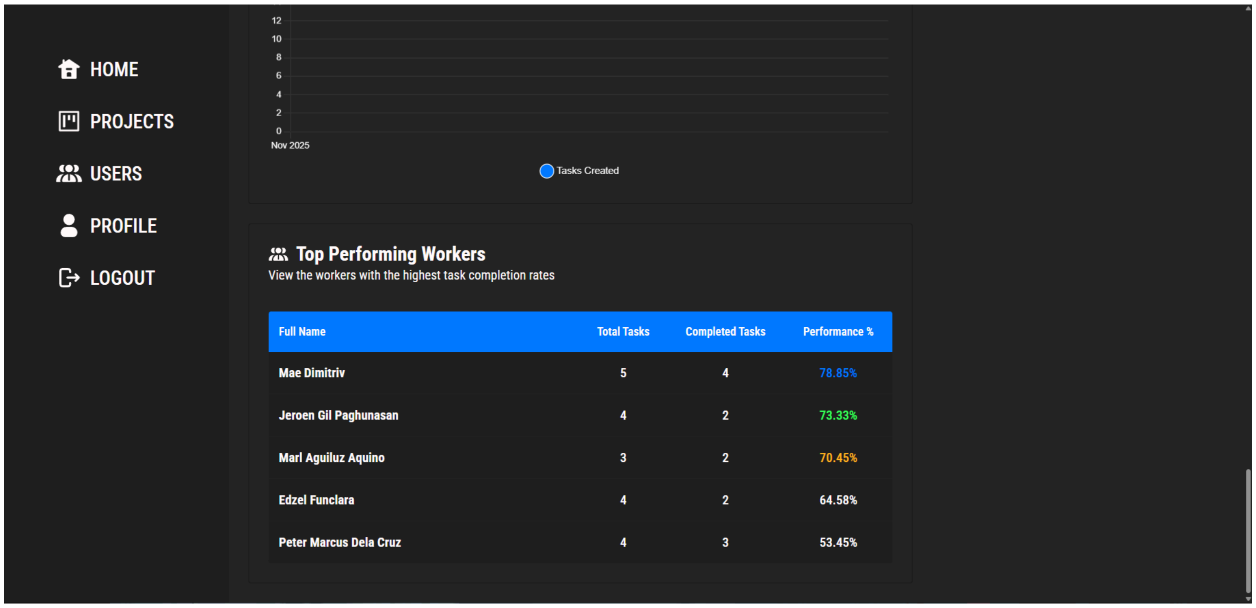
Tasks Created

Number of tasks created each period

Top Performing Workers

View the workers with the highest task completion rates

Full Name	Total Tasks	Completed Tasks	Performance %

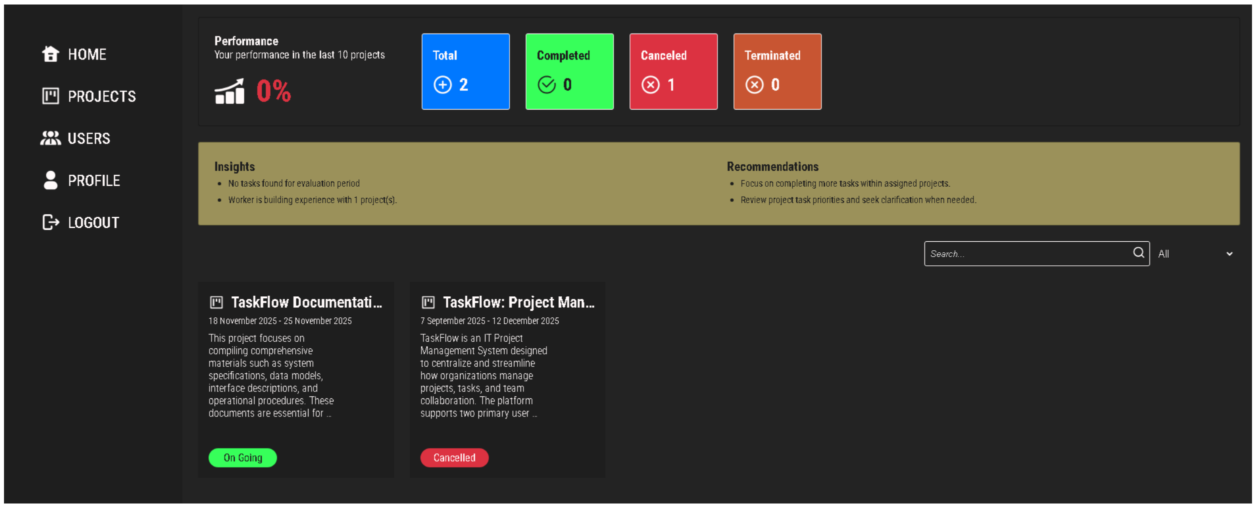


Top Performing Workers

View the workers with the highest task completion rates

Full Name	Total Tasks	Completed Tasks	Performance %
Mae Dimitriv	5	4	78.85%
Jeroen Gil Paghunasan	4	2	73.33%
Marl Aguiluz Aquino	3	2	70.45%
Edzel Funclara	4	2	64.58%
Peter Marcus Dela Cruz	4	3	53.45%

Figure 9. *Task Information Interface.*



Performance
Your performance in the last 10 projects
0%

Total	Completed	Canceled	Terminated
2	0	1	0

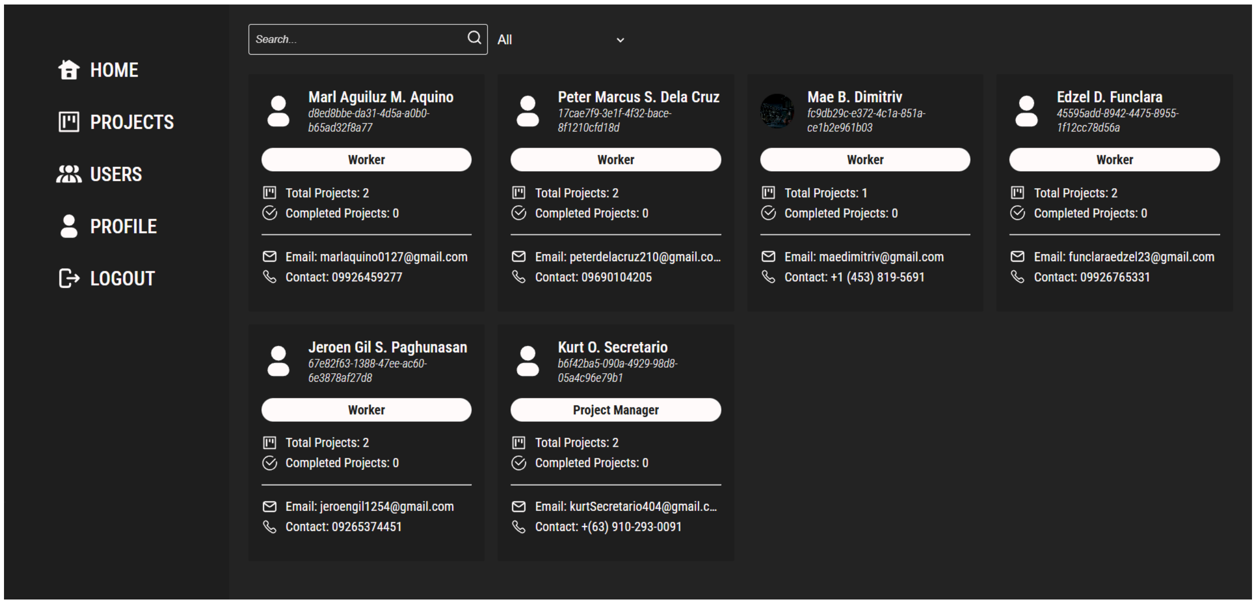
Insights
- No tasks found for evaluation period
- Worker is building experience with 1 project(s).

Recommendations
- Focus on completing more tasks within assigned projects.
- Review project task priorities and seek clarification when needed.

TaskFlow Documentati...
18 November 2025 - 25 November 2025
This project focuses on compiling comprehensive materials such as system specifications, data models, interface descriptions, and operational procedures. These documents are essential for ...
On Going

TaskFlow: Project Man...
7 September 2025 - 12 December 2025
TaskFlow is an IT Project Management System designed to centralize and streamline how organizations manage projects, tasks, and team collaboration. The platform supports two primary user ...
Cancelled

Figure 10. *Projects List Interface.*



Marl Aguiluz M. Aquino
d8ed8bbe-da31-4d5a-a0b0-b65ad32f8a77
Worker
Total Projects: 2
Completed Projects: 0
Email: marlaquino0127@gmail.com
Contact: 09926459277

Peter Marcus S. Dela Cruz
17cae7f9-3e1f-4f32-bace-8f1210cfd18d
Worker
Total Projects: 2
Completed Projects: 0
Email: peterdelacruz210@gmail.co...
Contact: 09690104205

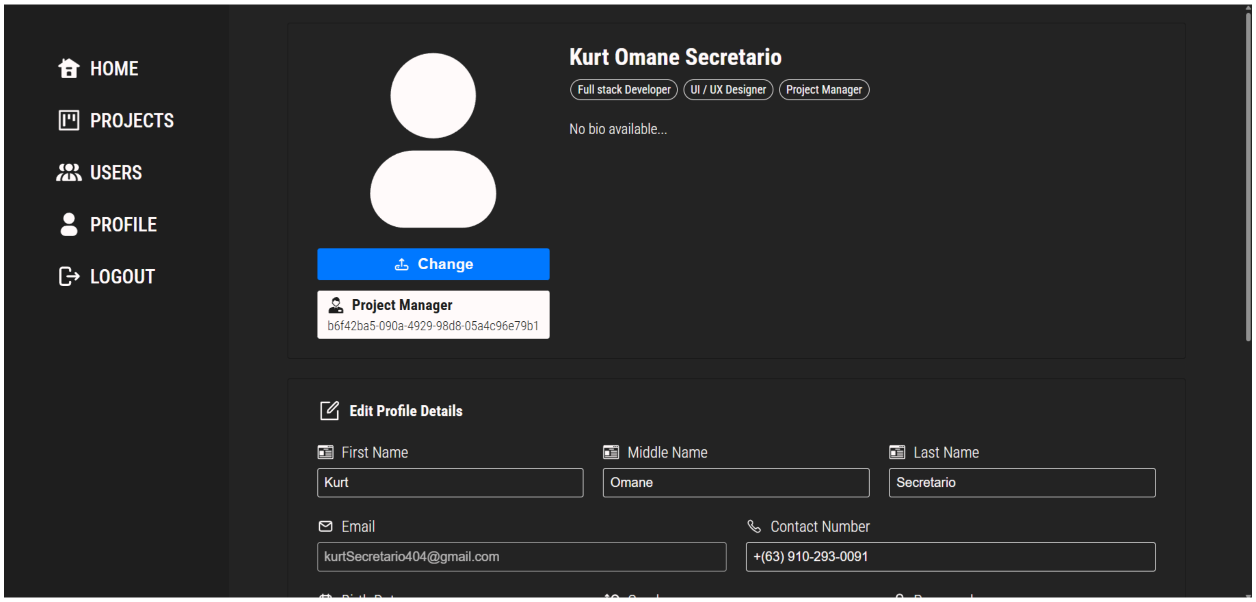
Mae B. Dimitriv
fc9db29c-e372-4c1a-851a-ce1b2e961b03
Worker
Total Projects: 1
Completed Projects: 0
Email: maedimitriv@gmail.com
Contact: +1 (453) 819-5691

Edzel D. Funclara
45595add-8942-4475-8955-1f12cc78d56a
Worker
Total Projects: 2
Completed Projects: 0
Email: funclaraedzel23@gmail.com
Contact: 09926765331

Jeroen Gil S. Paghunasan
67e82f63-1388-47ee-ac60-6e3878af27d8
Worker
Total Projects: 2
Completed Projects: 0
Email: jeroengil1254@gmail.com
Contact: 09265374451

Kurt O. Secretario
b6f42ba5-090a-4929-98d8-05a4c96e79b1
Project Manager
Total Projects: 2
Completed Projects: 0
Email: kurtSecretario404@gmail.c...
Contact: +(63) 910-293-0091

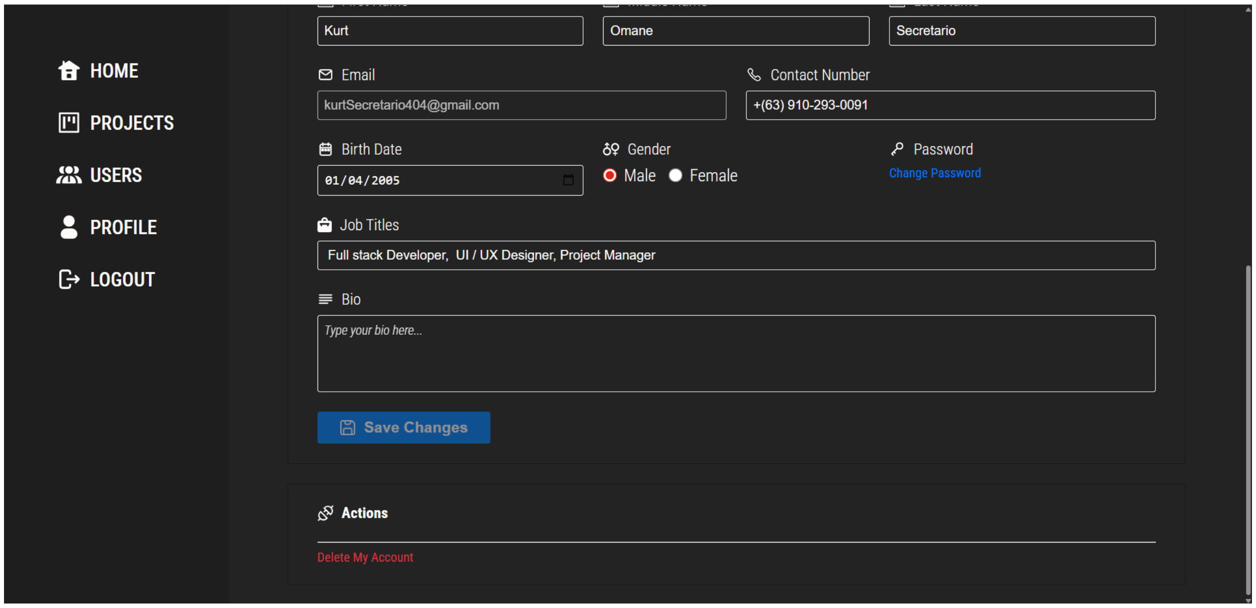
Figure 11. *Users Interface.*



Kurt Omane Secretario
Full stack Developer | UI / UX Designer | Project Manager
No bio available...

Project Manager
b6f42ba5-090a-4929-98d8-05a4c96e79b1

First Name: Kurt
Middle Name: Omane
Last Name: Secretario
Email: kurtSecretario404@gmail.com
Contact Number: +(63) 910-293-0091



Kurt | Omane | Secretario

Email: kurtSecretario404@gmail.com

Contact Number: +(63) 910-293-0091

Birth Date: 01/04/2005

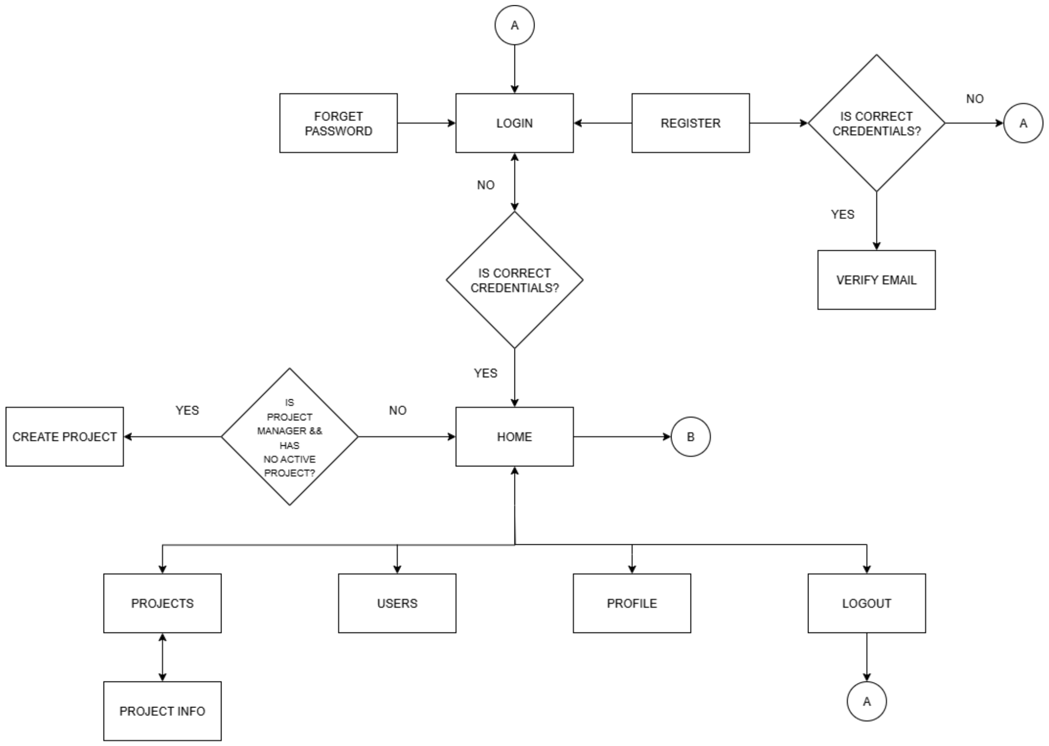
Gender: [✓ Male] Female

Job Titles: Full stack Developer, UI / UX Designer, Project Manager

Bio: *Type your bio here...*

Figure 12. *Profile Interface.*

Navigation Flow



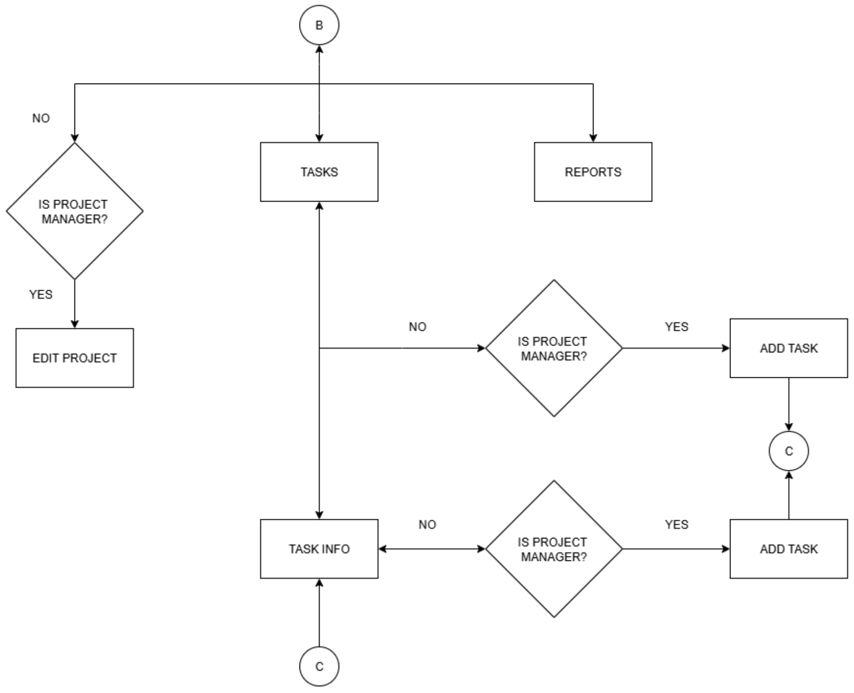


Figure 13. *Page navigation flow of TaskFlow.*

V. SYSTEM IMPLEMENTATION

Tools and Technologies Used

1. Backend Technologies

- PHP 8.2: Core server-side language leveraging modern features (enumerations, type hints, attributes, match expressions)
- MySQL/MariaDB 10.4+: Relational database with triggers, foreign keys, and CHECK constraints
- Composer: PHP dependency manager for third-party libraries

2. Backend Libraries/Frameworks

- PHPMailer: Email handling for registration confirmation, and password reset
- Cloudinary SDK: Cloud-based image storage and transformation for user profile pictures
- vlucas/phpdotenv: Environment variable management for configuration
- ramsey/uuid: UUID generation for public identifiers

3. Frontend Technologies

- HTML5: Semantic markup for views and components
- CSS3: Styling with custom properties, flexbox, grid layout (BEM-like naming convention)

- Vanilla JavaScript (ES6+): Client-side logic with modules, async/await, fetch API, arrow functions

4. Frontend Libraries

- Chart.js: Data visualization library for project progress charts, status distribution graphs, and performance metrics

5. Development Tools

- PNPM: Fast, disk-efficient JavaScript package manager
- Git: Version control system
- phpMyAdmin: Database management interface

6. Web Server

- Apache: HTTP server for routing and static asset serving
- mod_rewrite/.htaccess: URL rewriting for clean routes

7. Architecture Patterns & Conventions

- MVC Pattern: Custom implementation with Controllers, Models, Views, and Endpoints
- PSR-4 Autoloading: Namespace-based class autoloading via Composer
- RESTful API: HTTP methods (GET, POST, PUT, PATCH, DELETE) for CRUD operations

- JSON Configuration: Route definitions in JSON files for flexibility

Implementation Details

TaskFlow is a web-based hierarchical project management system designed to streamline task coordination, performance tracking, and progress reporting across organizational teams. The system manages a three-tier hierarchy (Projects → Phases → Tasks) with role-based access control for Project Managers and Workers. It features advanced performance calculation algorithms that evaluate worker productivity, manager effectiveness, and project progress using priority-weighted formulas. TaskFlow emphasizes transparency through documented mathematical metrics, automated status tracking via database triggers, and comprehensive visualization dashboards powered by Chart.js. The system supports collaborative task assignment, deadline management, and real-time performance insights to facilitate data-driven decision-making.

The system was developed using modern web technologies to ensure performance and maintainability. Its primary languages include PHP 8.2, JavaScript (ES6), SQL (MySQL/MariaDB), HTML5, and CSS3. Several backend tools were incorporated such as Composer, PHPMailer, and Cloudinary SDK, while the frontend uses Chart.js for visualizations and plain JavaScript for interactivity. Development relied on industry-standard tools, including Git/GitHub, VS Code, Apache/Nginx, and phpMyAdmin. Database operations were executed on a MySQL/MariaDB environment with full support for triggers, foreign keys, and constraints.

TaskFlow follows a custom MVC architectural pattern, consisting of several well-defined layers. The presentation layer handles HTML templates, components, JavaScript logic, and styling. Controllers and API endpoints form the application layer, managing routing, rendering, and JSON responses. The service layer contains business logic such as authentication workflows and performance calculators. The domain layer includes entities, validators, enums, and containers, ensuring consistency in data handling. Finally, the models in the data access layer handle CRUD operations through prepared SQL statements. Supporting this architecture is a JSON-based routing system and centralized path configuration to ensure clean URLs and maintainable file organization.

Each system module was implemented with clear workflows. The authentication module provides registration, login, email confirmation, and password reset, supported by session handling and CSRF protection. The task management module enables task creation, editing, assignment, searching, filtering, and status updating, with backend validation and database triggers ensuring correct workflows. The reporting module calculates project progress, worker performance scores, and manager efficiency using algorithm-based calculators and SQL queries. Visual representations of these metrics appear in the Chart.js-powered report pages. The dashboard module provides managers and workers with role-specific summaries, deadlines, activity logs, and quick actions.

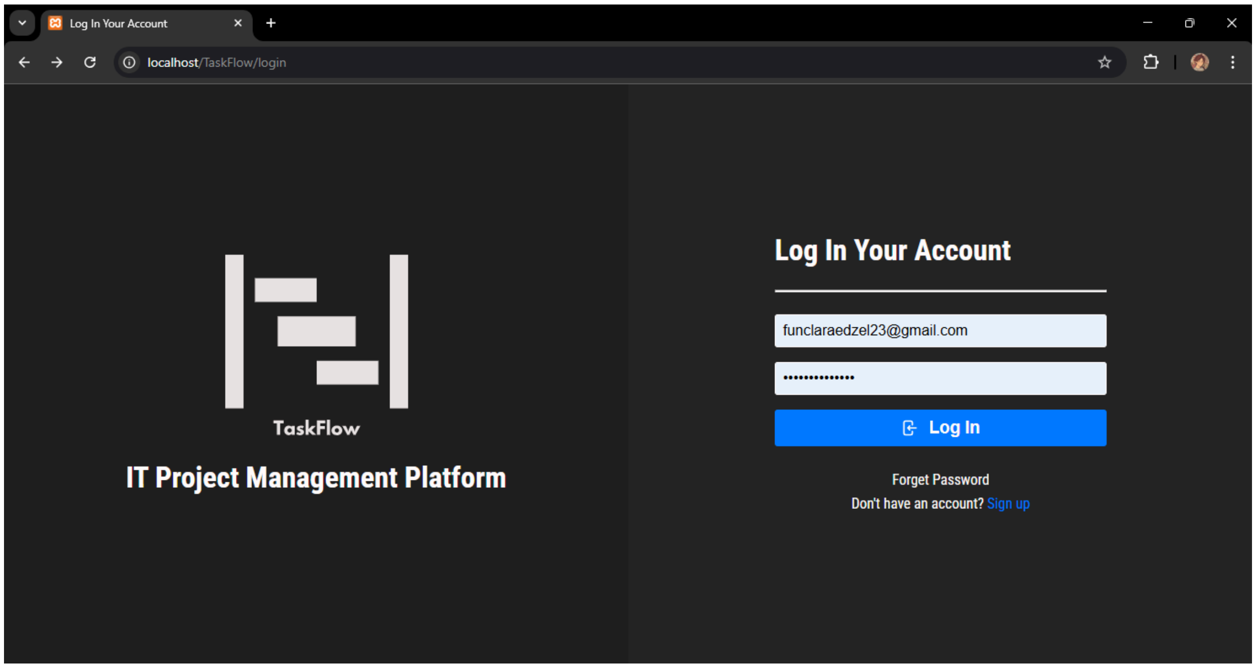
TaskFlow employs a structured relational database consisting of normalized tables for users, projects, phases, tasks, worker assignments, tokens, and job titles. Key relationships, such as one-to-many (project → phases) and many-to-many (tasks ↔

workers), are maintained through foreign keys. Several database triggers enforce business rules, including date validation, automatic completion timestamps, cascading cancellations, and user data anonymization. Indexes were created to improve query performance, especially for filtering and JOIN operations.

The system integrates external services to expand functionality. Email notifications, such as registration confirmation, password resets, task assignments, and deadline reminders, are handled through PHPMailer with SMTP transport. User profile images are stored and optimized via Cloudinary.

Security features were implemented across multiple layers. Passwords are protected using argon2id hashing, and all authenticated sessions follow strict rules such as token regeneration and expiration. Role-based access control ensures workers and managers can only access permitted actions. CSRF protection is applied to all sensitive requests, while input validation and prepared statements prevent injection attacks. Data privacy is further enforced using anonymization triggers, UUID-based public references, and HTTPS enforcement in production environments.

Sample Screenshots of Working System



Log In Your Account

funclaraedzel23@gmail.com

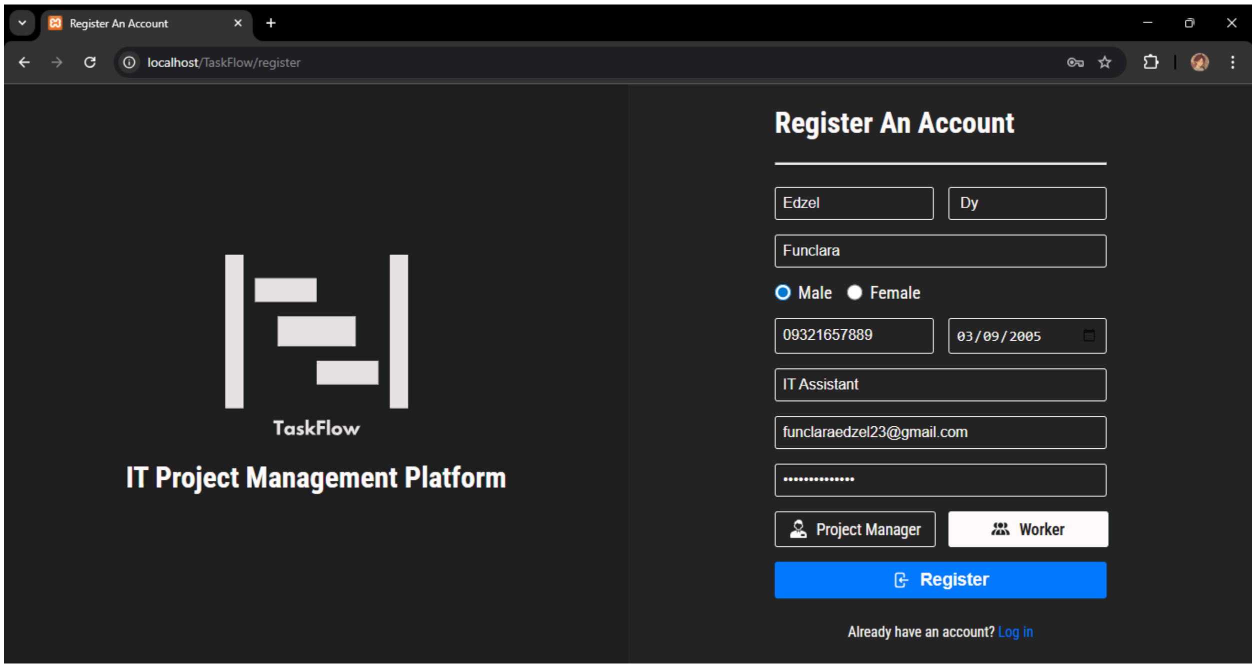
••••••••••••••

Log In

Forget Password

Don't have an account? Sign up

Figure 14. *Login interface of TaskFlow.*



Register An Account

Edzel | Dy

Funclara

[✓ Male] Female

09321657889 | 03/09/2005

IT Assistant

funclaraedzel23@gmail.com

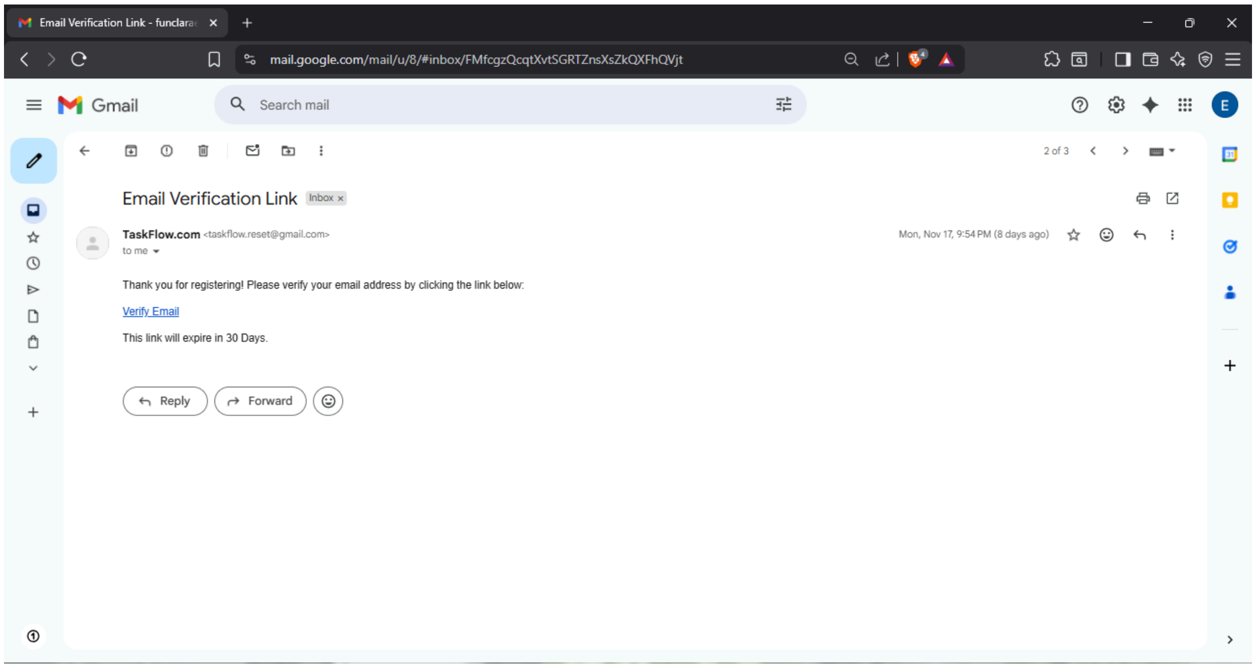
••••••••••••••

Project Manager | Worker

Register

Already have an account? Log in

Figure 15. *Registration for new users.*


Email Verification Link - funclara
mail.google.com/mail/u/8/#inbox/FMfcgzQcqtXvtSGRTZnsXsZkQXFhQVjt
Gmail
Search mail
2 of 3

Email Verification Link Inbox

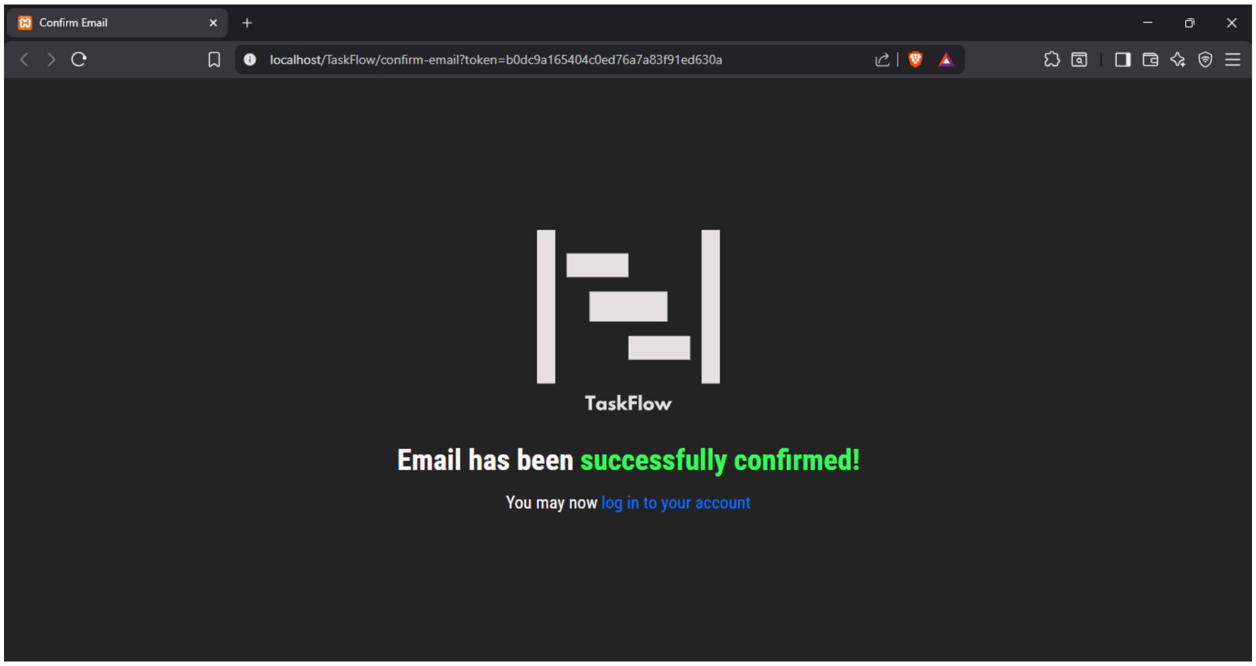
TaskFlow.com <taskflow.reset@gmail.com>
to me

Mon, Nov 17, 9:54 PM (8 days ago)

Thank you for registering! Please verify your email address by clicking the link below:

Verify Email

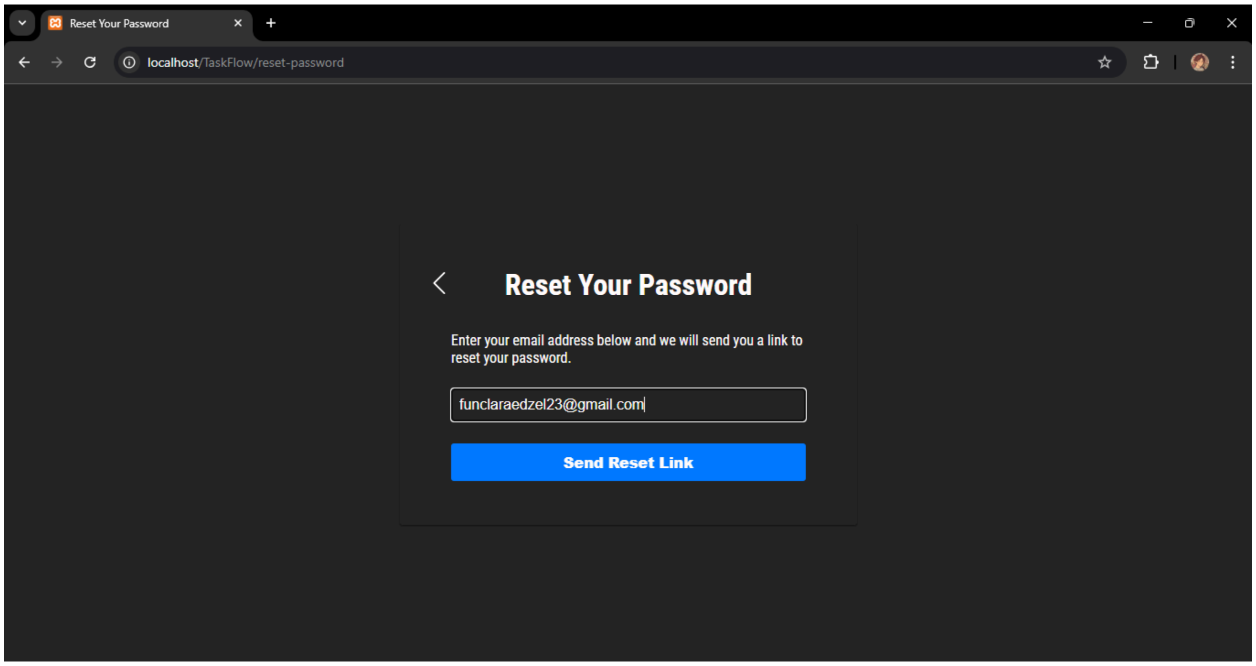
This link will expire in 30 Days.


Confirm Email
localhost/TaskFlow/confirm-email?token=b0dc9a165404c0ed76a7a83f91ed630a
TaskFlow

Email has been successfully confirmed!

You may now log in to your account

Figure 16. *Email verification for newly registered users.*

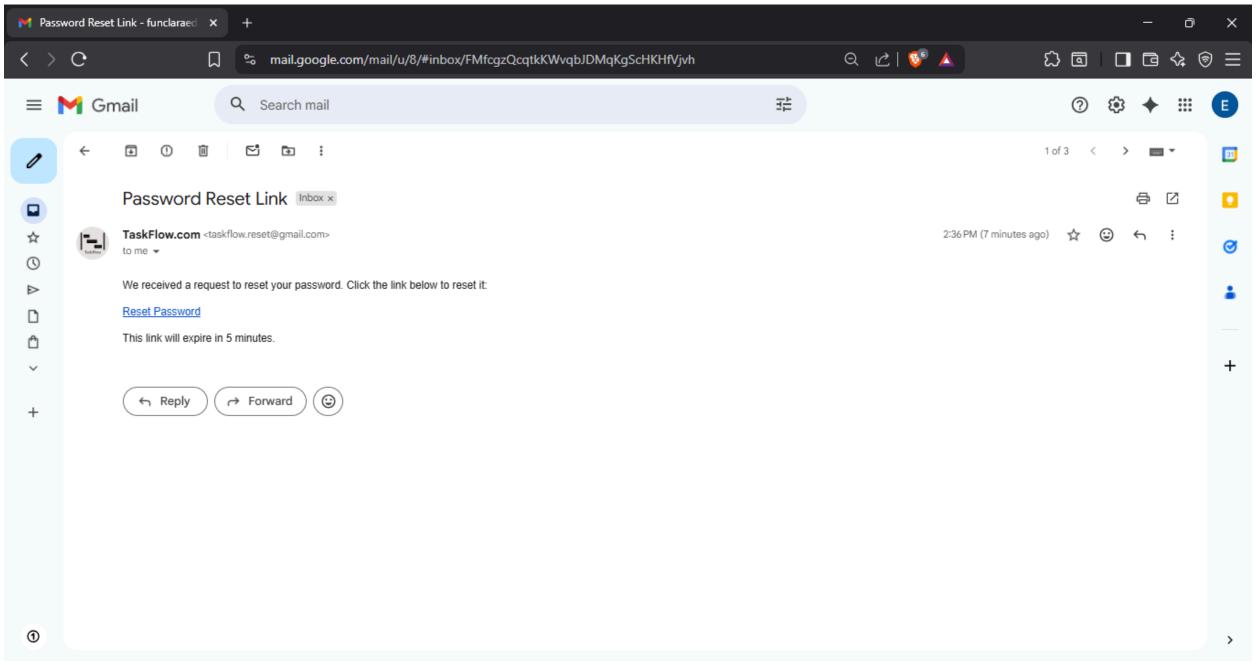


Reset Your Password

Enter your email address below and we will send you a link to reset your password.

funclaraedzel23@gmail.com

Figure 17. *Password reset for account recovery.*


Password Reset Link - funclaraed
mail.google.com/mail/u/8/#inbox/FMfcgzQcqtkKWvqbJDMqKgScHKHfVjvh
Gmail
Search mail
1 of 3
Inbox
2:36 PM (7 minutes ago)
Reply
Forward

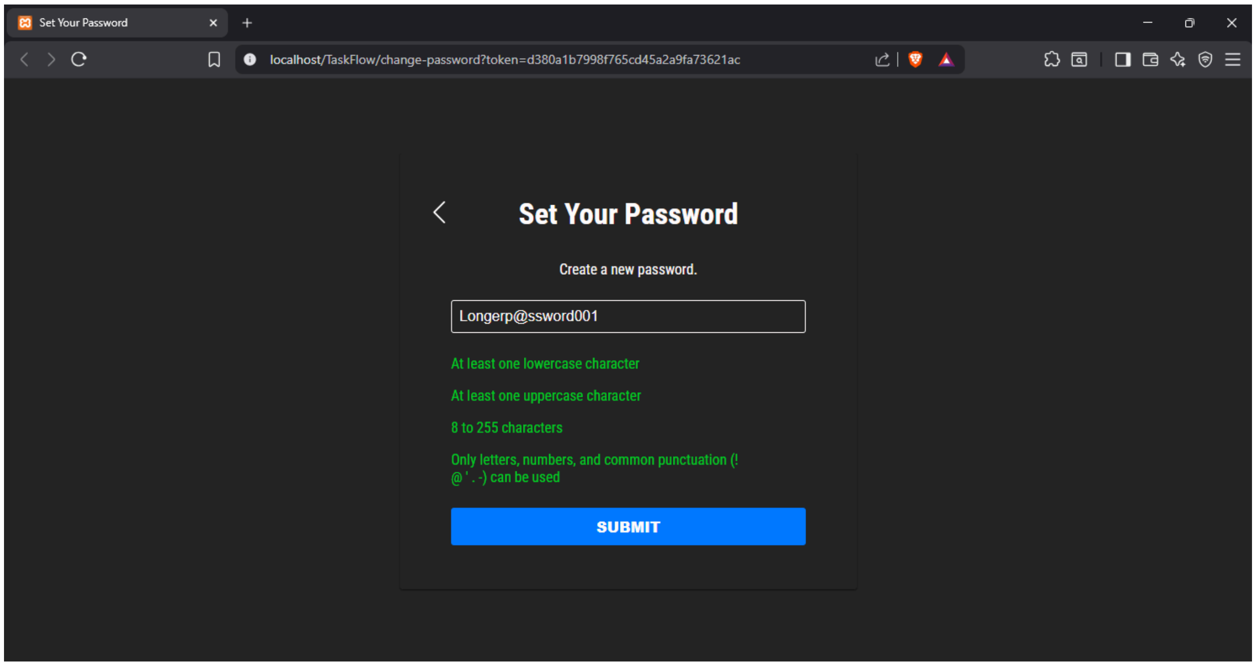
Password Reset Link

TaskFlow.com <taskflow.reset@gmail.com>
to me

We received a request to reset your password. Click the link below to reset it:

Reset Password

This link will expire in 5 minutes.


Set Your Password

localhost/TaskFlow/change-password?token=d380a1b7998f765cd45a2a9fa73621ac

Set Your Password

Create a new password.

Longerp@ssword001

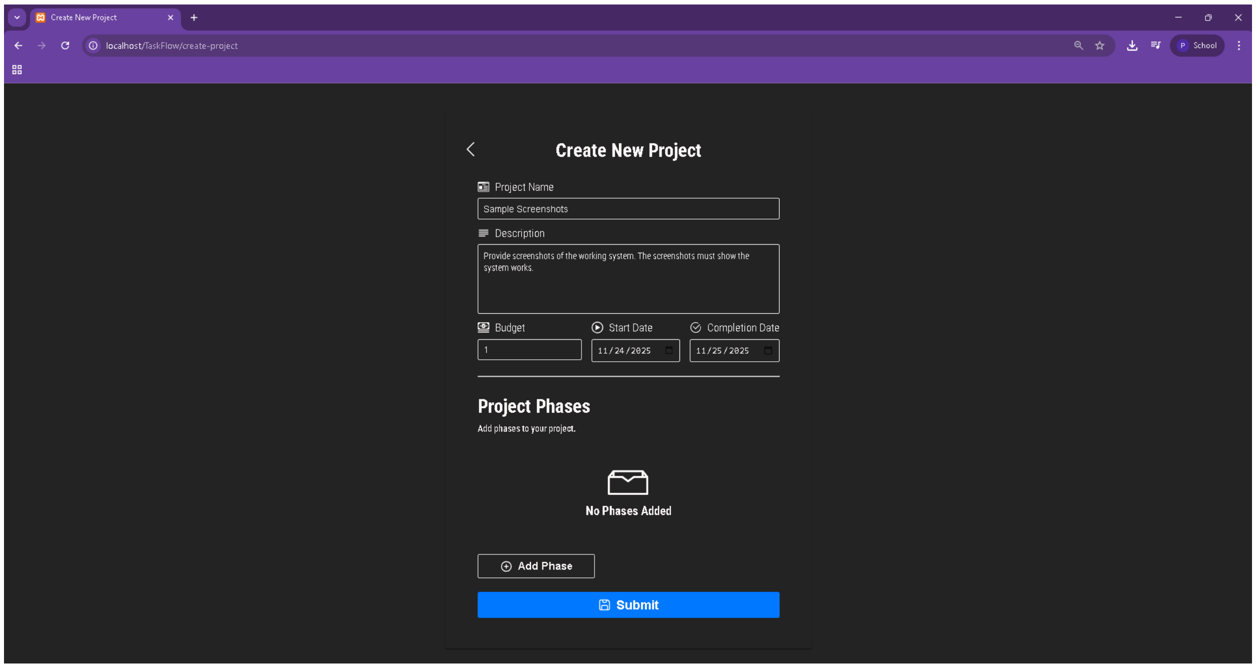
At least one lowercase character

At least one uppercase character

8 to 255 characters

Only letters, numbers, and common punctuation (! @ ' . -) can be used

Figure 18. *Password reset link sent via email.*


Create New Project

localhost/TaskFlow/create-project

Create New Project

Project Name: Sample Screenshots

Description: Provide screenshots of the working system. The screenshots must show the system works.

Budget: 1

Start Date: 11/24/2025

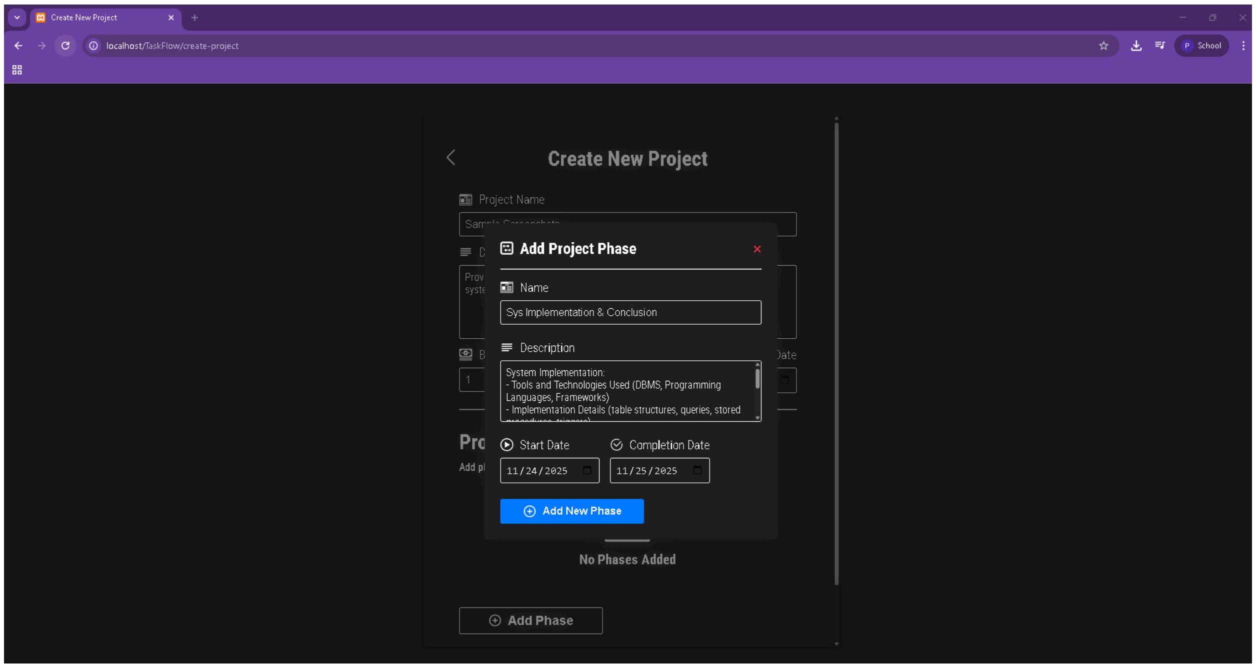
Completion Date: 11/25/2025

Project Phases

Add phases to your project.

No Phases Added

Figure 19. *Project creation form*.


Create New Project

Project Name

Add Project Phase

Name

Sys Implementation & Conclusion

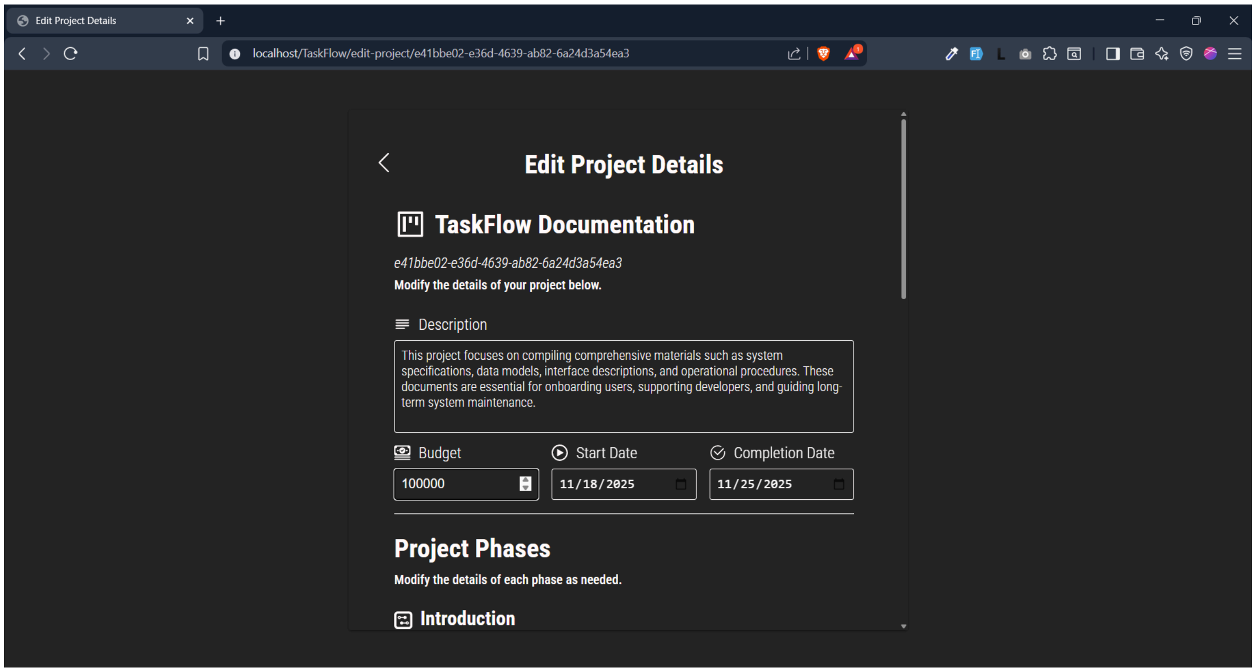
Description

System Implementation:
- Tools and Technologies Used (DBMS, Programming Languages, Frameworks)
- Implementation Details (table structures, queries, stored

Start Date 11/24/2025

Completion Date 11/25/2025

Figure 20. *Add a new phase to a project.*


localhost/TaskFlow/edit-project/e41bbe02-e36d-4639-ab82-6a24d3a54ea3

Edit Project Details

TaskFlow Documentation

e41bbe02-e36d-4639-ab82-6a24d3a54ea3

Modify the details of your project below.

Description

This project focuses on compiling comprehensive materials such as system specifications, data models, interface descriptions, and operational procedures. These documents are essential for onboarding users, supporting developers, and guiding long-term system maintenance.

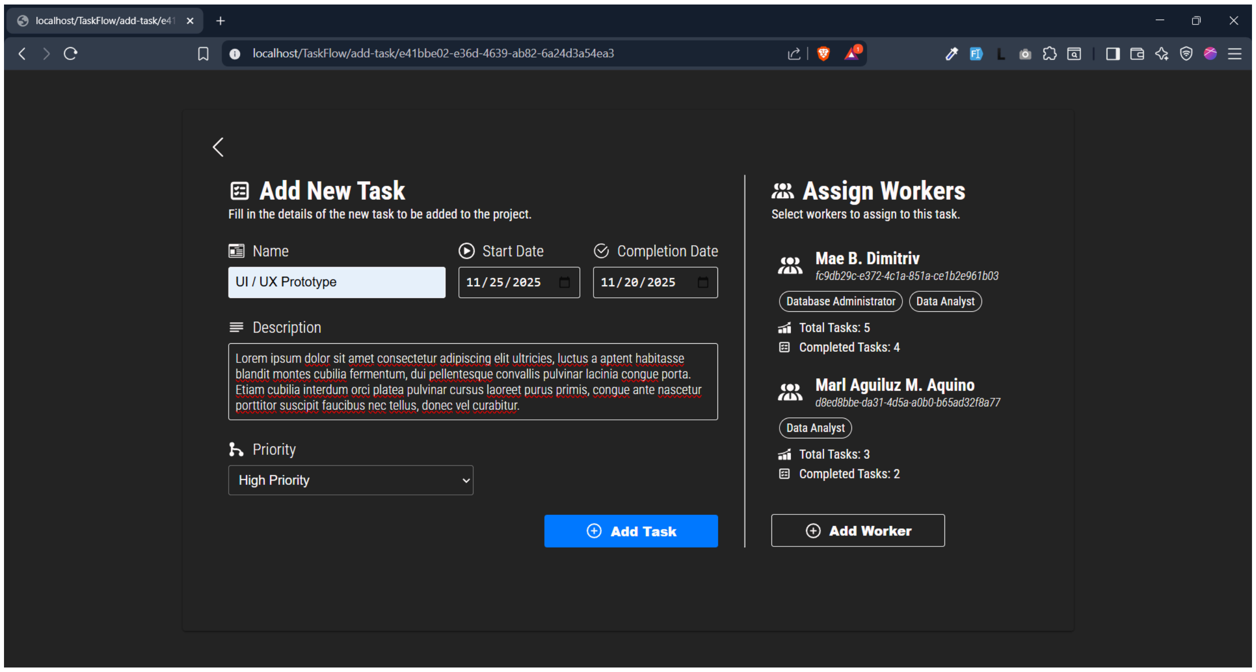
Budget	Start Date	Completion Date
100000	11/18/2025	11/25/2025

Project Phases

Modify the details of each phase as needed.

Introduction

Figure 21. *Edit project details.*


localhost/TaskFlow/add-task/e41bbe02-e36d-4639-ab82-6a24d3a54ea3

Add New Task

Fill in the details of the new task to be added to the project.

Name: UI / UX Prototype

Start Date: 11/25/2025

Completion Date: 11/20/2025

Description: Lorem ipsum dolor sit amet consectetur adipiscing elit ultricies, luctus a aptent habitasse blandit montes cubilia fermentum, dui pellentesque convallis pulvinar lacinia congue porta. Etiam cubilia interdum orci platea pulvinar cursus laoreet purus primis, congue ante nascetur porttitor suscipit faucibus nec tellus, donec vel curabitur.

Priority: High Priority

Add Task

Assign Workers

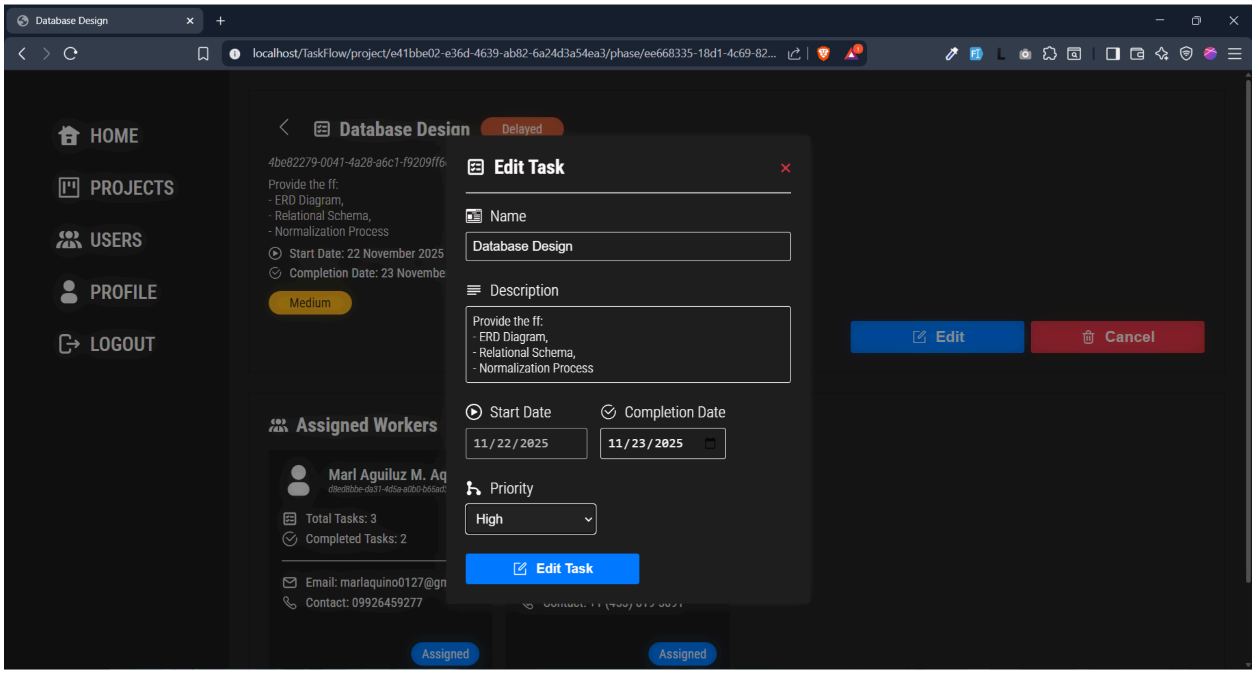
Select workers to assign to this task.

Mae B. Dimitriv
fc9db29c-e372-4c1a-851a-ce1b2e961b03
Database Administrator | Data Analyst
Total Tasks: 5
Completed Tasks: 4

Marl Aguiluz M. Aquino
d8ed8bbe-da31-4d5a-a0b0-b65ad32f8a77
Data Analyst
Total Tasks: 3
Completed Tasks: 2

Add Worker

Figure 22. *Add a new task to a project.*



Edit Task

Name: Database Design

Description:
Provide the ff:
- ERD Diagram,
- Relational Schema,
- Normalization Process

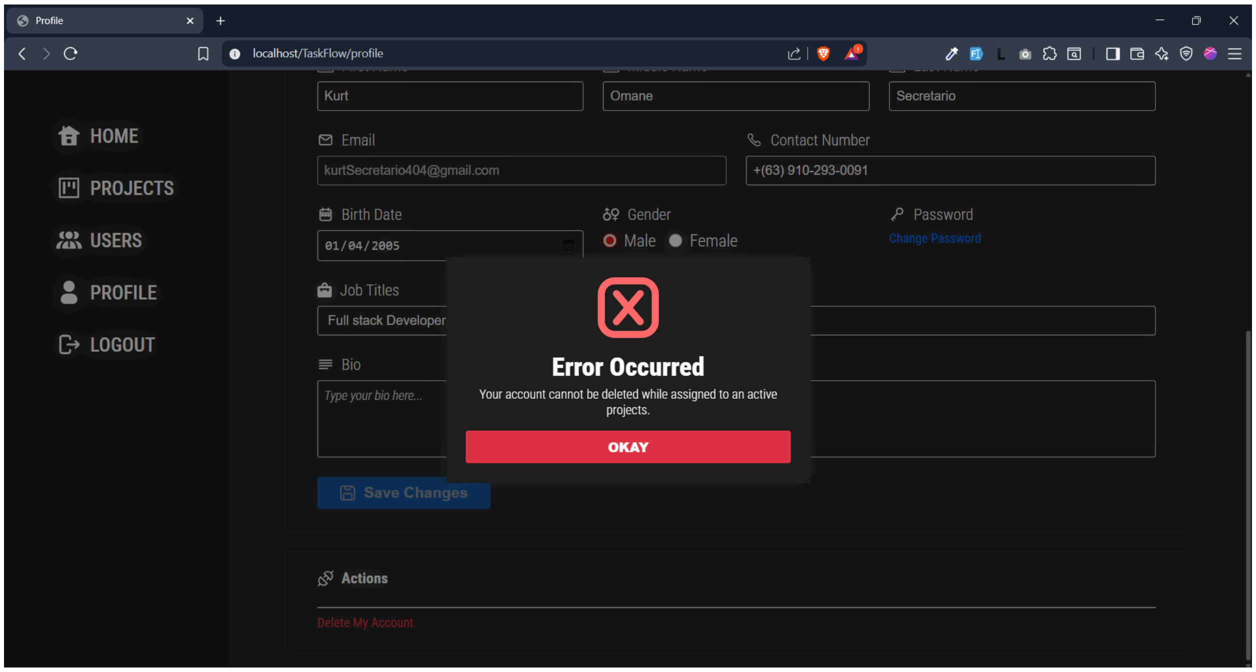
Start Date: 11/22/2025

Completion Date: 11/23/2025

Priority: High

Edit Task

Figure 23. *Edit task information.*



Kurt

Omane

Secretario

Email

kurtSecretario404@gmail.com

Contact Number

+(63) 910-293-0091

Birth Date

01/04/2005

Gender

Male Female

Password

Change Password

Job Titles

Full stack Developer

Bio

Type your bio here...

Save Changes

Actions

Delete My Account

Error Occurred

Your account cannot be deleted while assigned to an active projects.

OKAY

Figure 24. *Attempting to delete an account while an active project is ongoing.*

VII. CONCLUSIONS AND RECOMMENDATIONS

In conclusion, the development of the TaskFlow Project Management System offers significant improvements/benefits both the project management efficiency and performance outcomes. Having a user-friendly, customizable platform, TaskFlow enhances the ability to track and manage projects. Further, improving task completion rates and overall project timelines. Additionally, this has a big impact on project managers by boosting their performance through better monitoring tools that lead to higher project completion and improved time management. As for the Workers, TaskFlow's detailed task tracking and prioritization improve awareness of deadlines and encourage better engagement across multiple projects. Altogether, TaskFlow addresses the limitations of traditional project management methods, offering a more efficient and effective solution for managing complex workflows/projects.

In order for the effectiveness and real-world applicability of the system to be enhanced, it is recommended to apply multi-project management capabilities that can enable users to handle simultaneous projects more effectively. Combined with budget tracking, resource allocation monitoring, and real-time communication tools such, chat system, file sharing, and document co-editing. These recommendations would improve collaboration and offer a more real time project management experience. Additionally, applying advanced features such as automated workload balancing, predictive analytics for potential project delays, and task reassignment would help optimize project outcomes. To ensure data accuracy, the system could benefit from mechanisms to encourage timely and accurate updates. Lastly, while factors like behavior of employees and project interruptions are outside the system's scope, giving customizable options for teams to adapt the system to their unique needs will help lessen some of these outside factors.

VI. REFERENCES

Alves, P., Tereso, A., & Fernandes, G. (n.d.). *Project management system implementation in SMEs: A case study*. https://files.core.ac.uk/download/pdf/225349467.pdf

Aminu, N. H. (2024). *The role of technology in modern project management: Tools and techniques shaping the future*. *International Journal of Innovative Information Systems & Technology Research*.

Bloom, N., Liang, J., Roberts, J., & Ying, Z. J. (2015). Does working from home work? Evidence from a Chinese experiment. *The Quarterly Journal of Economics, 130*(1), 165–218. https://academic.oup.com/qje/article/130/1/165/2337855

Caniëls, M. C. J., & Bakens, R. J. J. M. (2012). The effects of project management information systems on decision making in a multi-project environment. *International Journal of Project Management, 30*(2), 162–175. https://doi.org/10.1016/j.ijproman.2011.06.005

Jain, A., Kolabkar, S., Netke, K., & Mate, P. (2020). Project Management System (PMS). *International Journal of Research in Engineering, Science and Management, 3*(6), 377–379. https://www.ijresm.com/Vol.3_2020/Vol3_Iss6_June20/IJRESM_V3_I6_98.pdf

Kretschmer, T., & Kude, T. (2021). Digital wellness in organizations: Measuring and enhancing well-being with wearable technology. *MIS Quarterly Executive, 20*(1), 51–66. https://misqe.org/ojs2/index.php/misqe/article/view/907

Maritim, K. T. (2022). *Project management information systems and decision-making in a multi-project environment* (Master's thesis). University of Nairobi.

Munyemana, B. (2022). *Project management information systems: An empirical study of their impact on project managers and project success*.

Mutwiri, I. M., & Muchelule, Y. (2022). *Project management information system integration and performance of public school construction projects in Kajiado County*. Research Publish.

VII. APPENDICES

1. Project full information query

```
SELECT
    p.*,
    JSON_OBJECT(
        'id', u.id,
        'publicId', HEX(u.publicId),
        'firstName', u.firstName,
        'middleName', u.middleName,
        'lastName', u.lastName,
        'email', u.email,
        'gender', u.gender,
        'profileLink', u.profileLink
    ) AS manager,
COALESCE(
    (
        SELECT CONCAT('[', GROUP_CONCAT(
            JSON_OBJECT(
                'id', pp.id,
                'publicId', HEX(pp.publicId),
                'name', pp.name,
                'description', pp.description,
                'startDateTime', pp.startDateTime,
                'completionDateTime', pp.completionDateTime,
                'actualCompletionDateTime', pp.actualCompletionDateTime,
                'status', pp.status,
                'createdAt', pp.createdAt
            ) ORDER BY pp.id ASC SEPARATOR ','
        ), ']')
        FROM `projectPhase` AS pp
        WHERE pp.projectId = p.id
    ), '[]'
) AS phases,
COALESCE(
    (
        SELECT CONCAT('[', GROUP_CONCAT(
            JSON_OBJECT(
                'workerId', w.id,
                'workerPublicId', HEX(w.publicId),
                'workerFirstName', w.firstName,
                'workerMiddleName', w.middleName,
                'workerLastName', w.lastName,
                'workerEmail', w.email,
                'workerGender', w.gender,
                'workerStatus', pw.status,
                'workerJobTitles', COALESCE(
                  (
                    SELECT
                      CONCAT('[', GROUP_CONCAT(CONCAT('\"', wjt.title, '\"')), ']')
                    FROM
                      `userJobTitle` AS wjt
                    WHERE
                      wjt.userId = w.id
                  ), '[]'
                )
            ) ORDER BY w.lastName ASC SEPARATOR ','
        ), ']')
        FROM `projectWorker` AS pw
        INNER JOIN `user` AS w ON pw.workerId = w.id
```

```
        WHERE pw.projectId = p.id

    ), '[]'

) AS workers,

COALESCE(

    (

        SELECT CONCAT('[', GROUP_CONCAT(

            JSON_OBJECT(

                'taskId', pt.id,

                'taskPublicId', HEX(pt.publicId),

                'taskName', pt.name,

                'taskDescription', pt.description,

                'taskPriority', pt.priority,

                'taskStatus', pt.status,

                'taskStartDateTime', pt.startDateTime,

                'taskCompletionDateTime', pt.completionDateTime,
                'taskActualCompletionDateTime', pt.actualCompletionDateTime
            ) ORDER BY pt.id ASC SEPARATOR ','
        ), ']')
        FROM `phaseTask` AS pt
        INNER JOIN `projectPhase` AS pp ON pt.phaseId = pp.id
        WHERE pp.projectId = p.id
    ), '[]'
  ) AS tasks
FROM `project` AS p
INNER JOIN `user` AS u ON p.managerId = u.id
WHERE p.id = :projectId
```

2. Phase information query

```
SELECT pp.* ,
  COALESCE(
    (
SELECT CONCAT('[', GROUP_CONCAT(
    JSON_OBJECT(
        'id', pt.id,
        'publicId', HEX(pt.publicId),
        'name', pt.name,
        'description', pt.description,
        'startDateTime', pt.startDateTime,
        'completionDateTime', pt.completionDateTime,
        'actualCompletionDateTime', pt.actualCompletionDateTime,
        'priority', pt.priority,
        'status', pt.status,
        'createdAt', pt.createdAt,
        'updatedAt', pt.updatedAt
    ) ORDER BY pt.id ASC SEPARATOR ','
), ']')
```

```
            FROM `phaseTask` AS pt
            WHERE pt.phaseId = pp.id
        ), '[]'
    ) AS tasks
FROM `projectPhase` AS pp
WHERE {dynamicWhereClause}
ORDER BY pp.id ASC
```

3. Task information query

```
SELECT
    pt.id AS taskId,
    pt.publicId AS taskPublicId,
    pp.publicId AS taskPhaseId,
    pt.name AS taskName,
    pt.description AS taskDescription,
    pt.startDateTime AS taskStartDateTime,
pt.completionDateTime AS taskCompletionDateTime,

pt.actualCompletionDateTime AS taskActualCompletionDateTime,

pt.priority AS taskPriority,

pt.status AS taskStatus,

pt.createdAt AS taskCreatedAt,

COALESCE(

    (

        SELECT CONCAT('[', GROUP_CONCAT(

            JSON_OBJECT(

                'workerId', u.id,

                'workerPublicId', HEX(u.publicId),

                'workerFirstName', u.firstName,

                'workerMiddleName', u.middleName,

                'workerLastName', u.lastName,

                'workerEmail', u.email,
        'workerContactNumber', u.contactNumber,

        'workerProfileLink', u.profileLink,

        'workerGender', u.gender,

        'workerStatus', ptw.status,

        'workerCreatedAt', u.createdAt,

        'workerConfirmedAt', u.confirmedAt,

        'workerDeletedAt', u.deletedAt,

        'workerJobTitles', COALESCE(

          (

               SELECT CONCAT('[', GROUP_CONCAT(CONCAT('\"', wjt.title, '\"')),
']')

             FROM `userJobTitle` AS wjt

             WHERE wjt.userId = u.id

          ), '[]'

        ),

        'workerTotalTasks', (
            SELECT COUNT(*)

            FROM `phaseTaskWorker` AS ptw2

            WHERE ptw2.workerId = u.id

        ),

        'workerCompletedTasks', (

            SELECT COUNT(*)

            FROM `phaseTaskWorker` AS ptw3

            INNER JOIN `phaseTask` AS pt3 ON ptw3.taskId = pt3.id

            WHERE ptw3.workerId = u.id

                AND pt3.status = 'completed'

        )

    ) ORDER BY u.lastName ASC SEPARATOR ','

), ']')

FROM `phaseTaskWorker` AS ptw

INNER JOIN `user` AS u ON ptw.workerId = u.id
```

```
            WHERE ptw.taskId = pt.id
        ), '[]'
    ) AS taskWorkers
FROM `phaseTask` AS pt
INNER JOIN `projectPhase` AS pp ON pt.phaseId = pp.id
INNER JOIN `project` AS p ON pp.projectId = p.id
LEFT JOIN `phaseTaskWorker` AS ptw ON pt.id = ptw.taskId
LEFT JOIN `user` AS u ON ptw.workerId = u.id
WHERE {dynamicWhereClause}
GROUP BY pt.id
ORDER BY pt.startDateTime DESC
LIMIT :limit OFFSET :offset
```

4. User information query

```
SELECT
    u.*,
GROUP_CONCAT(ujt.title) AS jobTitles,
(
    SELECT COUNT(DISTINCT p.id)
    FROM `project` AS p
    LEFT JOIN `projectWorker` AS pw ON p.id = pw.projectId
    WHERE p.managerId = u.id OR pw.workerId = u.id
) AS totalProjects,
(
    SELECT COUNT(DISTINCT p.id)
    FROM `project` AS p
    LEFT JOIN `projectWorker` AS pw ON p.id = pw.projectId
    WHERE (p.managerId = u.id OR pw.workerId = u.id)
        AND p.status = :completedStatus
) AS completedProjects,
(
        SELECT COUNT(DISTINCT p.id)

        FROM `project` AS p

        LEFT JOIN `projectWorker` AS pw ON p.id = pw.projectId

        WHERE pw.workerId = u.id

            AND p.status = :cancelledStatus

    ) AS cancelledProjectCount,

    (

        SELECT COUNT(*)

        FROM `projectWorker` AS pw

        WHERE pw.workerId = u.id

            AND pw.status = :terminatedStatus

    ) AS terminatedProjectCount

FROM `user` AS u

LEFT JOIN `userJobTitle` AS ujt ON u.id = ujt.userId

WHERE {dynamicWhereClause}
```

```
GROUP BY u.id

ORDER BY u.lastName ASC

LIMIT :limit OFFSET :offset
```

5. Temporary link query

```
SELECT *

FROM `temporaryLink`

WHERE token = :token

LIMIT 1
```